

RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	FFF	FFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU	NNN	NNN	000000000	FFF	FFF	
RRR	RRR	UUUUUUUUUUUUUUUU	NNN	NNN	000000000	FFF	FFF	
RRR	RRR	UUUUUUUUUUUUUUUU	NNN	NNN	000000000	FFF	FFF	

```
CCCCCCCC  AAAAAA  PPPPPPP  TTTTTTTTTT  IIIIII  000000  NN  NN
CCCCCCCC  AAAAAA  PPPPPPP  TTTTTTTTTT  IIIIII  000000  NN  NN
CC         AA      AA  PP      PP  TT      TT      II      II  00      00  NN  NN
CC         AA      AA  PP      PP  TT      TT      II      II  00      00  NN  NN
CC         AA      AA  PP      PP  TT      TT      II      II  00      00  NN  NN
CC         AA      AA  PP      PP  TT      TT      II      II  00      00  NN  NN
CC         AA      AA  PP      PP  TT      TT      II      II  00      00  NN  NN
CC         AA      AA  PP      PP  TT      TT      II      II  00      00  NN  NN
CC         AA      AA  PP      PP  TT      TT      II      II  00      00  NN  NN
CC         AA      AA  PP      PP  TT      TT      II      II  00      00  NN  NN
CC         AA      AA  PP      PP  TT      TT      II      II  00      00  NN  NN
CCCCCCCC  AA      AA  PP      PP  TT      TT      IIIIII  000000  NN  NN
CCCCCCCC  AA      AA  PP      PP  TT      TT      IIIIII  000000  NN  NN
```

```
LL         IIIIII  SSSSSSSS
LL         IIIIII  SSSSSSSS
LL         II      SS
LL         II      SS
LL         II      SS
LL         II      SS
LL         II      SSSSSS
LL         II      SSSSSS
LL         II      SS
LL         II      SS
LL         II      SS
LL         II      SS
LLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLL IIIIII  SSSSSSSS
```

.....

:

```
0001 0 %TITLE 'CAPTION - Generate figure, table and example captions'
0002 0 !<BLF/NOFORMAT>
0003 0
0004 0 MODULE caption (IDENT = 'V04-000'
0005 0 %BLISS32C, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE, NONEXTERNAL = LONG_RELATIVE)]
0006 0 ) =
0007 1 BEGIN
0008 1
0009 1 !<BLF/FORMAT>
0010 1 !<BLF/LOWERCASE_USER>
0011 1 !<BLF/UPPERCASE_KEY>
0012 1 !<BLF/MACRO>
0013 1
0014 1
0015 1 *****
0016 1 *
0017 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0018 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0019 1 * ALL RIGHTS RESERVED.
0020 1 *
0021 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0022 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0023 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0024 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0025 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0026 1 * TRANSFERRED.
0027 1 *
0028 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0029 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0030 1 * CORPORATION.
0031 1 *
0032 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0033 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0034 1 *
0035 1 *
0036 1 *****
0037 1
0038 1
0039 1 ++
0040 1 FACILITY:
0041 1 DSR (Digital Standard RUNOFF) /DSRPLUS DSRTOC/CONTENTS Utility
0042 1
0043 1 ABSTRACT:
0044 1 Format figure, table and example captions
0045 1
0046 1 ENVIRONMENT: Transportable
0047 1
0048 1 AUTHOR: JPK
0049 1
0050 1 CREATION DATE: March 1982
0051 1
0052 1 MODIFIED BY:
0053 1
0054 1 007 JPK00010 05-May-1983
0055 1 Modified CONTENTS to change text for index record group to all
0056 1 caps. Modified CAPTION to change the number of blank lines
0057 1 after the entity title line.
```



```
58 0058 1
59 0059 1
60 0060 1
61 0061 1
62 0062 1
63 0063 1
64 0064 1
65 0065 1
66 0066 1
67 0067 1
68 0068 1
69 0069 1
70 0070 1
71 0071 1
72 0072 1
73 0073 1
74 0074 1
75 0075 1
76 0076 1
77 0077 1
78 0078 1
79 0079 1
80 0080 1
81 0081 1
82 0082 1
83 0083 1
84 0084 1
85 0085 1
86 0086 1
87 0087 1
88 0088 1
89 0089 1
90 0090 1
91 0091 1
92 0092 1
93 0093 1
94 0094 1
95 0095 1
96 0096 1
97 0097 1
98 0098 1
99 0099 1
100 0100 1
101 0101 1
102 0102 1
103 0103 1
104 0104 1
105 0105 1
106 0106 1
107 0107 1
108 0108 1
109 0109 1
110 0110 1
111 0111 1
112 0112 1
113 0193 1
114 0194 1
```

006 JPK00008 09-Mar-1983
Modified CONTENTS and CAPTION to support new BRN formats,
support SEND CONTENTS, /DOUBLE_SPACE, page numbered chapters,
guarantee space after section number and to write new
prologue and epilog for RUNOFF output.
Modified FORMAT to quote only the RUNOFF flags used by CONTENTS.
Modified CNTVMS to fix default for /DOUBLE_SPACE and do more
value checking.

005 JPK00007 14-Feb-1983
Global edit of all sources for CONTENTS/DSRTOC:
- module names are now consistent with file names
- copyright dates have been updated
- facility names have been updated
- revision history was updated to be consistent with DSR/DSRPLUS

004 JPK00004 11-Feb-1983
Changed the global variable name INDENT to LINE_INDENT in
modules CONTENTS, CAPTION, FORMAT and GBLDCL.
Removed declarations of PDENTS in modules CNTVMS, CONTENTS,
and CAPTION and replaced with a module wide BIND using the
new name INDENTS.
Changed handling of INDENTS [1]. It no longer represents the
sum of the chapter and title indents.

003 KFA00002 14-Oct-1982
Modified CNTVMS, CONTENTS, CAPTION and CNTCLI.REQ to handle
new syntax for /INDENT qualifier. All indents are now
stored in the vector CMDBLK [CONTENTSSAG_HL_INDENT []],
where 0 = chapter indent, 1 = header level one indent, etc.

002 KFA00001 13-Oct-1982
Modified CONTENTS and CAPTION to allow null captions for
headers, examples, figures, & tables.

--

TABLE OF CONTENTS:

FORWARD ROUTINE

hdr_x : NOVALUE, ! Process Figure, Table and Example captions
stcHdr : NOVALUE, ! Process SEND TOC temporary file
strip : NOVALUE, ! Remove leading and trailing whitespace
insrnt : NOVALUE, ! Insert text directly into output file

INCLUDE FILES:

LIBRARY 'NXPORT:XPORT';

REQUIRE 'REQ:PAGEN'; ! Page reference format

REQUIRE 'REQ:TOCRTY'; ! Table of Contents file formats

```
115 0304 1
116 0305 1 REQUIRE 'REQ:CNCL1';
117 0433 1 ! Command line information block formats
118 0434 1
119 0435 1 ! MACROS:
120 0436 1
121 0437 1
122 0438 1 MACRO
123 0439 1
124 0440 1 ! Write a character to output line
125 0441 1
126 M 0442 1 write_char (ch) [] =
127 M 0443 1 BEGIN
128 M 0444 1 CH$WCHAR_A (ch, lp);
129 M 0445 1 intlin = .intlin + 1;
130 M 0446 1
131 M 0447 1 %IF NOT %NULL (%REMAINING)
132 M 0448 1 %THEN
133 M 0449 1 extlin = .extlin + 1;
134 M 0450 1 %FI
135 M 0451 1
136 M 0452 1 END
137 0453 1
138 0454 1 !
139 0455 1 ! Write a text literal to the output line
140 0456 1
141 M 0457 1 literal_text (str) =
142 M 0458 1 BEGIN
143 M 0459 1 CH$MOVE (%CHARCOUNT (str), CH$PTR (UPLIT (str)), .lp);
144 M 0460 1 lp = CH$PLUS (.lp, %CHARCOUNT (str));
145 M 0461 1 intlin = .intlin + %CHARCOUNT (str);
146 M 0462 1 END
147 0463 1 !
148 0464 1 !
149 0465 1 ! Pad the output line with blanks
150 0466 1
151 M 0467 1 pad (n_blanks) =
152 M 0468 1 BEGIN
153 M 0469 1
154 M 0470 1 IF n_blanks GTR 0
155 M 0471 1 THEN
156 M 0472 1 BEGIN
157 M 0473 1 CH$FILL (%C' ', n_blanks, .lp);
158 M 0474 1 lp = CH$PLUS (.lp, n_blanks);
159 M 0475 1 intlin = .intlin + n_blanks;
160 M 0476 1 extlin = .extlin + n_blanks;
161 M 0477 1 END;
162 M 0478 1
163 M 0479 1 END
164 0480 1 !
165 0481 1 !
166 0482 1 ! Clear the text lines being built up.
167 0483 1
168 M 0484 1 clr_line (_) =
169 M 0485 1 BEGIN
170 M 0486 1 lp = CH$PTR (line);
171 M 0487 1 intlin = 0;
```



```
172 M 0488 1      extlin = 0;
173 M 0489 1      END
174      0490 1      |
175      0491 1      |
176      0492 1      | Insert specified character sequence into file, as is.
177      0493 1      |
178 M 0494 1      put (str) =
179 M 0495 1      BEGIN
180 M 0496 1      $str_copy (string = str, target = tmpstr);
181 M 0497 1      chROUT = .chROUT + .tmpstr [str$.length];
182 M 0498 1      $xpo_put (iob = tocoob, string = tmpstr);
183 M 0499 1      |
184 M 0500 1      | For TMS output, split the output file if it gets too large
185 M 0501 1      |
186 M 0502 1      |
187 M 0503 1      IF .cndblk [contents$v_tms11] THEN split ();
188 M 0504 1      |
189 M 0505 1      END
190      0506 1      |
191      0507 1      |
192      0508 1      |
193      0509 1      | EQUATED SYMBOLS:
194      0510 1      |
195      0511 1      |
196      0512 1      LITERAL
197      0513 1      true = 1
198      0514 1      false = 0
199      0515 1      tab = %0'11' : UNSIGNED (8),
200      0516 1      rintex = %0'34' : UNSIGNED (8),
201      0517 1      infinity = 1^(%BPVAL - 1) - 1;
202      0518 1      |
203      0519 1      |
204      0520 1      | OWN STORAGE:
205      0521 1      |
206      0522 1      |
207      0523 1      |
208      0524 1      |
209      0525 1      |
210      0526 1      EXTERNAL
211      0527 1      cndblk : $contents_cmd,
212      0528 1      tocoob : $xpo_iob (?),
213      0529 1      stciob : $xpo_iob (?),
214      0530 1      chROUT,
215      0531 1      tmpstr : $str_descriptor (),
216      0532 1      txtlen,
217      0533 1      hl_n,
218      0534 1      hlList : VECTOR [100],
219      0535 1      hldsp : VECTOR [100],
220      0536 1      major,
221      0537 1      minor,
222      0538 1      pagen : page_definition,
223      0539 1      nextstc,
224      0540 1      rno_ex : $str_descriptor (),
225      0541 1      rno_fg : $str_descriptor (),
226      0542 1      rno_tb : $str_descriptor (),
227      0543 1      tms_ex : $str_descriptor (),
228      0544 1      tms_fg : $str_descriptor (),

Command line information block
IOB for the resulting .RNT file
IOB for send toc temp file
Number of characters written to output file
Temporary strings
Length of input text.
"n" from latest .HL n command
Header level counters
Header level display characteristics
Major record type code
Minor record type code
Current page number.
Next SEND TOC "n" value to process
String constant 'EXAMPLES'
String constant 'FIGURES'
String constant 'TABLES'
String constant 'Examples'
String constant 'Figures'
```

```
229 0545 1 tms_tb : $str_descriptor (),      | String constant 'Tables'
230 0546 1 prestr : $str_descriptor (),      | Caption prefix
231 0547 1 sfxstr : $str_descriptor (),      | Caption suffix
232 0548 1 lp,                                | CHSPTR along line being built up
233 0549 1 intlin,                            | Number of characters needed to represent text
234 0550 1 extlin,                            | Number of resulting print positions
235 0551 1 line : VECTOR [CH$ALLOCATION (10000)], | Buffer in which line is being built up.
236 0552 1 lenpag,                            | Number of characters in the converted page number.
237 0553 1 txtpag : VECTOR [CH$ALLOCATION (50)], | The text (lots of room)
238 0554 1 ptrpag,                            | A CHSPTR (working pointer)
239 0555 1 lensct,                            | Number of characters in the converted section number
240 0556 1 txtsct : VECTOR [CH$ALLOCATION (150)], | The text (lots of room)
241 0557 1 ptrsct,                            | A CHSPTR (working pointer)
242 0558 1 rmargin,                          | Used by ENDWRD for controlling filling lines.
243 0559 1 wrap,                            | Wrap long lines around to here.
244 0560 1 line_indent;                      | Assume this standard indentation before the text.
245 0561 1
246 0562 1 BIND
247 0563 1     indents = cmdblk[contents$ag_hl_indent] : VECTOR;
248 0564 1
249 0565 1 EXTERNAL ROUTINE
250 0566 1     split : NOVALUE,                | Start new output file for TMS
251 0567 1     fmttxt : NOVALUE,              | Generate formatted text
252 0568 1     insref : NOVALUE,              | Insert a page reference
253 0569 1     pacpag,                        | Format a page number
254 0570 1     pacbas,                        | Convert a number to ASCII
255 0571 1     pacsec,                        | Format a section number
256 0572 1     pacxxx;                       | Format a number using display code
257 0573 1
258 L 0574 1 %IF %BLISS (BLISS32)
259 0575 1 %THEN
260 0576 1
261 0577 1 EXTERNAL ROUTINE
262 0578 1     open_error;                    | File open error handling routine
263 0579 1
264 0580 1 %FI
```


CAPTION
V04-000

CAPTION - Generate figure, table and example ca
HDR_X - Process .HEADER x captions

M 9
15-Sep-1984 23:50:23
14-Sep-1984 13:05:37

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]CAPTION.BLI;1

Page 6
(2)

CAP1
V04-

```
266 0581 1 %SBTTL 'HDR X - Process .HEADER x captions'
267 0582 1 GLOBAL ROUTINE hdr_x (
268 0583 1     iob : REF $xpo_iob (),
269 0584 1     title : REF $str_descriptor (),
270 0585 1     info,
271 0586 1     txt,
272 0587 1     txt_u
273 0588 1 ) : NOVALUE =
274 0589 1 ++
275 0590 1
276 0591 1 FUNCTIONAL DESCRIPTION:
277 0592 1
278 0593 1     This routine processes the .HEADER x Captions
279 0594 1
280 0595 1 FORMAL PARAMETERS:
281 0596 1
282 0597 1     iob      - address of temp file iob
283 0598 1     title   - address of string descriptor of title
284 0599 1     info    - information record type
285 0600 1     txt     - text record type
286 0601 1     txt_u   - unnumbered text record type
287 0602 1
288 0603 1 IMPLICIT INPUTS:
289 0604 1
290 0605 1     cmdblk  - command line information block
291 0606 1
292 0607 1 IMPLICIT OUTPUTS:
293 0608 1
294 0609 1     None
295 0610 1
296 0611 1 ROUTINE VALUE:
297 0612 1 COMPLETION CODES:
298 0613 1
299 0614 1     None
300 0615 1
301 0616 1 SIDE EFFECTS:
302 0617 1
303 0618 1     None
304 0619 1
305 0620 1 --
306 0621 1
307 0622 1 BEGIN
308 0623 1
309 0624 1 LOCAL
310 0625 1     first_record;
311 0626 1
312 0627 1 IF .cmdblk [contents$g_headers] EQL 0 THEN RETURN;
313 0628 1
314 0629 1 first_record = true;
315 P 0630 1 $xpo_open (iob = .iob, options = input, attributes = BINARY
316 P 0631 1     %IF %BLISS (BLISS32) %THEN , failure = open_error %FI
317 0632 1 );
318 0633 1
319 0634 1 WHILE $xpo_get (iob = .iob, fullwords = 2) EQL xpo$_normal DO
320 0635 1     BEGIN
321 0636 1     ! Process a record group
322 0637 1
```



```
323 0638 !
324 0639 ! BIND
325 0640 record_header = .iob [iob$a_data] : VECTOR;
326 0641
327 0642 major = .record_header [0];
328 0643 minor = .record_header [1];
329 0644
330 0645 SELECTONE true OF
331 0646 SET
332 0647
333 0648 [(.major EQL maj_runoff) AND (.minor EQL min_page)] :
334 0649 BEGIN
335 0650 ! Process a page reference
336 0651 !
337 0652 ! $xpo_get (iob = .iob, fullwords = page_sct_size);
338 0653 BEGIN
339 0654 BIND
340 0655 page_ref = .iob [iob$a_data] : VECTOR;
341 0656
342 0657
343 0658 MAP
344 0659 pagen : VECTOR;
345 0660
346 0661 INCR i FROM 0 TO page_sct_size - 1 DO
347 0662 pagen [i] = .page_ref [i];
348 0663
349 0664 END;
350 0665 END;
351 0666
352 0667 [.major EQL maj_send_contents] :
353 0668 BEGIN
354 0669 ! Process a SEND CONTENTS record
355 0670 !
356 0671 ! LOCAL
357 0672 ! type;
358 0673 !
359 0674 ! type = .minor<0, %BPVAL/2>;
360 0675 ! $xpo_get (iob = .iob, fullwords = 1); ! Get length
361 0676 ! txtlen = ..iob [iob$a_data];
362 0677 !
363 0678 ! IF .txtlen GTR 0
364 0679 ! THEN
365 0680 ! $xpo_get (iob = .iob, fullwords = CH$ALLOCATION (.txtlen));
366 0681 !
367 0682 ! IF .type EQL scon_comm
368 0683 ! THEN
369 0684 ! ! A command. Insert directly into table of contents
370 0685 ! ! put ((.txtlen, CH$PTR (.iob [iob$a_data])))
371 0686 ! !
372 0687 ! ! ELSE
373 0688 ! ! BEGIN
374 0689 ! ! Text. Format it.
375 0690 ! ! IF .first_record
376 0691 ! !
377 0692 ! !
378 0693 ! !
379 0694 ! !
```

```
380 0695 5 THEN
381 0696 6 BEGIN
382 0697 6
383 0698 6 First one seen - generate title
384 0699 6
385 0700 6 IF .cmdblk [contents$V_tms11]
386 0701 6 THEN
387 0702 6 BEGIN
388 0703 6 put ('');
389 0704 6
390 0705 6 IF .hl_n NEQ 0 THEN put ('[va15]');
391 0706 6
392 0707 6 put ($str_concat ('*c0n*', .title, '@@'));
393 0708 6 END
394 0709 6 ELSE
395 0710 6 BEGIN
396 0711 6 put ('.TEST PAGE 7');
397 0712 6 IF .hl_n NEQ 0 THEN put ('.SK2') ELSE put ('.SK1');
398 0713 6
399 0714 6 IF .indents [0] NEQ 0
400 0715 6 THEN
401 0716 6 put ($str_concat ('.I-', $str_ascii (.indents [0])));
402 0717 6
403 0718 6 put (,title);
404 0719 6 put ('.BLANK');
405 0720 6 END;
406 0721 6
407 0722 6 hl_n = 1;
408 0723 6 first_record = false;
409 0724 6 END;
410 0725 6
411 0726 6
412 0727 6 Initialize formatting parameters
413 0728 6
414 0729 6 clr_line ();
415 0730 6 line_indent = .indents [0];
416 0731 6 extlin = .line_indent;
417 0732 6 wrap = .indents [0] + .indents [1];
418 0733 6 rmargin = .cmdblk [contents$g_page_width];
419 0734 6
420 0735 6 IF .cmdblk [contents$V_tms11]
421 0736 6 THEN
422 0737 6 literal_text ('*c1n*@')
423 0738 6 ELSE
424 0739 6 pad ((.indents [0] + .indents [1] - .extlin));
425 0740 6
426 0741 6 IF .type EQL scon_txt
427 0742 6 THEN
428 0743 6 BEGIN
429 0744 6
430 0745 6 Generate the page number so as to get its length.
431 0746 6 This is needed so that the proper number of dots can be supplied.
432 0747 6
433 0748 6 ptrpag = CH$PTR (txtpag);
434 0749 6
435 0750 6 IF NOT .cmdblk [contents$V_standard_page]
436 0751 6 THEN
```

```

lenpag = pacbas (.pagen [sct_run_page], ptrpag, 10)
ELSE
lenpag = pacpag (pagen, ptrpag);

END
ELSE
lenpag = 0;

| Set limit beyond which words must be wrapped
rmargin = .cmdblk [contents$g_page_width] - .lenpag - 1;
wrap = .extlin;

| Scan text(if any) and put it(may be null caption) onto the line.
fmttxt (.txtlen, CH$PTR (.iob [iob$a_data]));

IF .type EQL scon_txt
THEN
insref ()
! Insert page reference
ELSE
BEGIN
IF .cmdblk [contents$y_tms11] THEN write_char (XC'a');
END;

| Now actually put this line of text into the file
put ((.intlin, CH$PTR (line)));
clr_line ();
END;

END;

[(.major EQL maj_runoff) AND (.minor EQL .info)] :
BEGIN
| A caption information record
LOCAL
prefix_len,
suffix_len;

| Get caption counter, display descriptor and prefix text length
$spo get (iob = .iob, fullwords = 3);
BEGIN
BIND
info_vector = .iob [iob$a_data] : VECTOR [3];

hllist [0] = .info_vector [0]; ! Save the counter
hldsp [0] = .info_vector [1]; ! the display descriptor
prefix_len = .info_vector [2]; ! and prefix string length
```



```
END;
Get prefix text if any
IF .prefix_len NEQ 0 THEN $xpo_get (iob = .iob, fullwords = CH$ALLOCATION (.prefix_len));
$str_copy (string = (.prefix_len, CH$PTR (.iob [iob$a_data])), target = prestr);
Get length of suffix text
$xpo_get (iob = .iob, fullwords = 1);
suffix_len = (.iob [iob$a_data]);
Get suffix text if any
IF .suffix_len NEQ 0 THEN $xpo_get (iob = .iob, fullwords = CH$ALLOCATION (.suffix_len));
$str_copy (string = (.suffix_len, CH$PTR (.iob [iob$a_data])), target = sfxstr);
END;
[(.major EQL maj_runoff) AND (.minor EQL .txt OR .minor EQL .txt_u)] :
BEGIN
Process the text part of a caption.
LOCAL
needed_fill;

First read the number of characters needed to represent the caption
$xpo_get (iob = .iob, fullwords = 1);
txtlen = (.iob [iob$a_data]); ! Save text length.
Generate a table of contents entry even if the caption
has no text associated with it
If .txtlen NEQ 0
THEN
$xpo_get (iob = .iob, fullwords = CH$ALLOCATION (.txtlen));

Initialize formatting parameters
clr_line ();
line_indent = .indents [0];
extlin = .line_indent;
wrap = .indents [0] + .indents [1];
rmargin = .cmdblk [contents$g_page_width];
IF .first_record
THEN
BEGIN
First one seen - generate title
IF .cmdblk [contents$v_tms11]
```

```
551 0866 5 THEN
552 0867 6 BEGIN
553 0868 6 put ('');
554 0869 6
555 0870 6 IF .hl_n NEQ 0 THEN put ('[va15]');
556 0871 6
557 0872 6 put ($str_concat ('*c0n*', .title, 'aa'));
558 0873 6 END
559 0874 5 ELSE
560 0875 6 BEGIN
561 0876 6 put ('.TEST PAGE 7');
562 0877 6 IF .hl_n NEQ 0 THEN put ('.SK2') ELSE put ('.SK1');
563 0878 6
564 0879 6 IF .indents [0] NEQ 0
565 0880 6 THEN
566 0881 6 put ($str_concat ('.I-', $str_ascii (.indents [0])));
567 0882 6
568 0883 6 put (',title);
569 0884 6 put (',.BLANK');
570 0885 6 END;
571 0886 5
572 0887 5 hl_n = 1;
573 0888 5 first_record = false;
574 0889 5 END;
575 0890 4
576 0891 4 IF .cmdblk [contents$u_tms11] THEN literal_text ('*c1n*');
577 0892 4
578 0893 4 IF .cmdblk [contents$u_include_sections]
579 0894 5 AND (
580 0895 6 (.cmdblk [contents$g_sections] GEQ .hl_n)
581 0896 6 OR (
582 0897 7 (.cmdblk [contents$g_sections] EQL -1)
583 0898 7 AND (.minor NEQ .txt_u)
584 0899 6 )
585 0900 5 )
586 0901 4 THEN
587 0902 5 BEGIN
588 0903 6
589 0904 6 Generate number if either the user said include
590 0905 6 all header numbers up to some level GEQ 1 or
591 0906 6 header numbers should be displayed as input and
592 0907 6 this was a numbered head.
593 0908 6
594 0909 6 ptrsct = CHSPTR (txtsct);
595 0910 6 lensct = 0;
596 0911 6
597 0912 6 IF .pagen [sct_typ] NEQ 0
598 0913 6 THEN
599 0914 7 BEGIN
600 0915 8
601 0916 8 Document is divided into sections
602 0917 8
603 0918 8 lensct = .lensct + pacsec (pagen, ptrsct) + 1;
604 0919 8 CHSWCHAR_A (%C'-', ptrsct);
605 0920 8 END;
606 0921 7
607 0922 5 lensct = .lensct + pacxxx (.hllist [0], ptrsct, .hldsp [0]);
```

```
608 0923 5
609 0924 5
610 0925 5
611 0926 5
612 0927 5
613 0928 5
614 0929 5
615 0930 5
616 0931 5
617 0932 5
618 0933 5
619 0934 5
620 0935 5
621 0936 5
622 0937 5
623 0938 5
624 0939 6
625 0940 6
626 0941 6
627 0942 6
628 0943 6
629 0944 6
630 0945 6
631 0946 5
632 0947 5
633 0948 5
634 0949 4
635 0950 4
636 0951 4
637 0952 4
638 0953 5
639 0954 4
640 0955 4
641 0956 4
642 0957 4
643 0958 4
644 0959 4
645 0960 4
646 0961 4
647 0962 4
648 0963 4
649 0964 4
650 0965 5
651 0966 5
652 0967 5
653 0968 5
654 0969 5
655 0970 5
656 0971 4
657 0972 5
658 0973 5
659 0974 5
660 0975 5
661 0976 5
662 0977 4
663 0978 4
664 0979 4

Write the prefix string if any
IF .prestr [str$h_length] NEQ 0
THEN
    fmttxt (.prestr [str$h_length], .prestr [str$a_pointer]);

Write the entity number
fmttxt (.lensct, CH$PTR (txtsct));

Write the suffix string if any
IF .sfxstr [str$h_length] NEQ 0
THEN
    BEGIN
        IF CH$NEQ(1, CH$PTR (UPLIT(XC' '))),
            .sfxstr [str$h_length], .sfxstr [str$a_pointer])
        THEN
            fmttxt (.sfxstr [str$h_length], .sfxstr [str$a_pointer]);
    END;

IF NOT .cmdblk [contents$v_tms11] THEN write_char (XC' ', counts_visually);
END;

IF .cmdblk [contents$v_tms11]
THEN
    write_char (XC'@')
ELSE
    pad ((.indents [0] + .indents [1] - .extlin));

Generate the page number so as to get its length.
This is needed so that the proper number of dots can be supplied.
ptrpag = CH$PTR (txtpag);

IF NOT .cmdblk [contents$v_standard_page]
THEN
    BEGIN
        Use the running page counter instead of a real page number.
        lenpag = pacbas (.pagen [sct_run_page], ptrpag, 10);
    END
ELSE
    BEGIN
        Supply the usual page number.
        lenpag = pacpag (pagen, ptrpag);
    END;
END;
```



```
665 0980 4      | Set limit beyond which words must be wrapped
666 0981 4
667 0982 4      rmargin = .cndblk [contents$g_page_width] - .lenpag - 1;
668 0983 4      wrap = .extlin;
669 0984 4
670 0985 4      | Scan text(if any) and put it(may be null caption) onto the line.
671 0986 4      ftxtxt (.txtlen, CHSPTR (.iob [iob$a_data]));
672 0987 4
673 0988 4      | Insert page reference into line if desired.
674 0989 4      insref ();
675 0990 4
676 0991 4      | Now actually put this line of text into the file
677 0992 4      put ((.intlin, CHSPTR (line)));
678 0993 4      clr_line ();
679 0994 4      END;
680 0995 4
681 0996 4      TES;
682 0997 4
683 0998 4      END;
684 0999 4
685 1000 4      $xpo_close (iob = .iob, options = remember);
686 1001 4      END;
687 1002 4
688 1003 4
689 1004 4
```

```
37 20 45 47 41 50 20 54 53 45 54 2E 00000 P.AAA: .BLKB 0
2A 6E 30 63 2A 00000 P.AAB: .ASCII \[va15]\
00006 P.AAE: .ASCII \*c0n*\
0000B P.AAF: .ASCII \aa\
0000D P.AAI: .ASCII \.TEST PAGE 7\
32 4B 53 2E 00019 P.AAJ: .ASCII \.SK2\
31 4B 53 2E 0001D P.AAK: .ASCII \.SK1\
2D 49 2E 00021 P.AAM: .ASCII \.I-\
4B 4E 41 4C 42 2E 00024 P.AAO: .ASCII \.BLANK\
0002A .BLKB 2
00 00 40 2A 6E 31 63 2A 0002C P.AAP: .ASCII \*c1n*a\<0><0>
5D 35 31 61 76 5B 00034 P.AAQ: .BLKB 0
2A 6E 30 63 2A 00034 P.AAR: .ASCII \[va15]\
0003A P.AAU: .ASCII \*c0n*\
37 20 45 47 41 50 20 54 53 45 54 2E 0003F P.AAV: .ASCII \aa\
32 4B 53 2E 00041 P.AAY: .ASCII \.TEST PAGE 7\
31 4B 53 2E 0004D P.AAZ: .ASCII \.SK2\
2D 49 2E 00051 P.ABA: .ASCII \.SK1\
4B 4E 41 4C 42 2E 00055 P.ABC: .ASCII \.I-\
00058 P.ABE: .ASCII \.BLANK\
0005E .BLKB 2
00 00 00 2A 6E 31 63 2A 00060 P.ABF: .ASCII \*c1n*\<0><0><0>
0000003A 00068 P.ABG: .LONG 58
```

```
.TITLE CAPTION CAPTION - Generate figure, table and ex
.IDENT \V04-000\
.PSECT $SPLITS,NOWRT,NOEXE,2
```

2. NO EXE. SOWNS. PSECT

0000	00000	\$STR\$STRING:	
01 0E	00002	.WORD	0
00000000	00004	.BYTE	14, 1
0006	00008	.ADDRESS	P.AAA
01 0E	0000A	\$STR\$STRING:	
00000000	0000C	.WORD	6
0005	00010	.BYTE	14, 1
01 0E	00012	.ADDRESS	P.AAB
00000000	00014	\$STR\$STRING:	
0002	00018	.WORD	5
01 0E	0001A	.BYTE	14, 1
00000000	0001C	.ADDRESS	P.AAE
000C	00020	\$STR\$STRING:	
01 0E	00022	.WORD	2
00000000	00024	.BYTE	14, 1
0004	00028	.ADDRESS	P.AAF
01 0E	0002A	\$STR\$STRING:	
00000000	0002C	.WORD	12
0004	00030	.BYTE	14, 1
01 0E	00032	.ADDRESS	P.AAI
00000000	00034	\$STR\$STRING:	
0003	00038	.WORD	4
01 0E	0003A	.BYTE	14, 1
00000000	0003C	.ADDRESS	P.AAJ
0006	00040	\$STR\$STRING:	
01 0E	00042	.WORD	4
00000000	00044	.BYTE	14, 1
0000	00048	.ADDRESS	P.AAK
01 0E	0004A	\$STR\$STRING:	
00000000	0004C	.WORD	3
0006	00050	.BYTE	14, 1
01 0E	00052	.ADDRESS	P.AAM
00000000	00054	\$STR\$STRING:	
0005	00058	.WORD	6
01 0E	0005A	.BYTE	14, 1
00000000	0005C	.ADDRESS	P.AAO
0002	00060	\$STR\$STRING:	
01 0E	00062	.WORD	0
00000000	00064	.BYTE	14, 1
000C	00068	.ADDRESS	P.AAQ
01 0E	0006A	\$STR\$STRING:	
		.WORD	6
		.BYTE	14, 1
		.ADDRESS	P.AAR
		\$STR\$STRING:	
		.WORD	5
		.BYTE	14, 1
		.ADDRESS	P.AAU
		\$STR\$STRING:	
		.WORD	2
		.BYTE	14, 1
		.ADDRESS	P.AAV
		\$STR\$STRING:	
		.WORD	12
		.BYTE	14, 1

```
00000000' 0006C .ADDRESS P.AAY
0004 00070 $STR$STRING:
01 0E 00072 .WORD 4
00006000' 00074 .BYTE 14, 1
0004 00078 $STR$STRING:
01 0E 0007A .WORD 4
00000000' 0007C .BYTE 14, 1
0003 00080 $STR$STRING:
01 0E 00082 .WORD 3
00000000' 00084 .BYTE 14, 1
0006 00088 $STR$STRING:
01 0E 0008A .WORD 6
00000000' 0008C .BYTE 14, 1
. ADDRESS P.ABE
```

```
.EXTRN CMDBLK, TOCOOB, STCI0B
.EXTRN CHROUT, TMPSTR, TXTLEN
.EXTRN HL_N, HLLIST, HLDSP
.EXTRN MAJOR, MINOR, PAGEN
.EXTRN NXTSTC, RNO_EX, RNO_FG
.EXTRN RNO_TB, TMS_EX, TMS_FG
.EXTRN TMS_TB, PRESTR, SFXSTR
.EXTRN LP, INLIN, EXLIN
.EXTRN LINE, LENPAG, TXTPAG
.EXTRN PTRPAG, LENSCT, TXTSCT
.EXTRN PTRSCT, RMARGIN
.EXTRN WRAP, LINE INDENT
.EXTRN SPLIT, FMTXT, INSREF
.EXTRN PACPAG, PACBAS, PACSEC
.EXTRN PACXXX, OPEN ERROR
.EXTRN XPOSOPEN, XPOSGET
.EXTRN XPOSFAILURE, XST$COPY
.EXTRN STR$FAILURE, XPOSPUT
.EXTRN XST$JOIN, XST$ASCII
.EXTRN XPOS$CLOSE
```

```
.PSECT $CODE$,NOWRT,2
```

```
5B 00000000G EF 9E 00002 .ENTRY HDR_X, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 : 0582
5E 00000000G 08 C2 00009 MOVAB $STR$TARGET, R11
01 12 00012 SUBL2 #8, SP
04 00014 TSTL CMDBLK+4 : 0627
01 D0 00015 RET 1$
04 00018 MOVL #1, FIRST_RECORD : 0629
01 AC D0 00018 MOVL IOB, R6 : 0632
2E A6 00010001 8F C8 0001C BISL2 #65537, 46(R6)
2C A6 00000000G 01 90 00024 MOVB #1, 44(R6)
04 00028 PUSHAB OPEN ERROR
7E D4 0002E CLRL -(SP)
56 DD 00030 PUSHL R6
03 FB 00032 CALLS #3, XPOSOPEN
34 38 A6 9E 00039 MOVAB 56(R6), R8 : 0640
08 B0 0003D 2$ MOVW #8, 52(R6) : 0634
```


36	A6	02	90	00041	MOV	#2, 54(R6)	
2C	A6	06	90	00045	MOV	#6, 44(R6)	
		EF	9F	00049	PUSHAB	XPOSFAILURE	
		7E	D4	0004F	CLRL	-(SP)	
		56	DD	00051	PUSHL	R6	
00000000G	EF	03	FB	00053	CALLS	#3, XPOSGET	
00208001	8F	50	D1	0005A	CMPL	R0, #2129921	
		03	13	00061	BEQL	3\$	
		OCF0	31	00063	BRW	74\$	
		68	D0	00066	MOVL	(R8), R0	0640
00000000G	EF	60	D0	00069	MOVL	(R0), MAJOR	0642
00000000G	EF	04	A0	00070	MOVL	4(R0), MINOR	0643
		51	D4	00078	CLRL	R1	0648
		02	00000000G	EF	D1	0007A	
		02	12	00081	CMPL	MAJOR, #2	
		51	D6	00083	BNEQ	4\$	
		50	D4	00085	INCL	R1	
		01	00000000G	EF	D1	00087	
		02	12	0008E	CLRL	R0	
		50	D6	00090	CMPL	MINOR, #1	
		51	D2	00092	BNEQ	5\$	
		52	CA	00095	INCL	R0	
		01	D1	00098	MCOML	R1, R2	
		30	12	0009B	BICL2	R2, R0	
34	A6	10	B0	0009D	CMPL	R0, #1	
36	A6	02	90	000A1	BNEQ	7\$	
2C	A6	06	90	000A5	MOVW	#16, 52(R6)	0653
		EF	9F	000A9	MOV	#2, 54(R6)	
		7E	D4	000AF	MOV	#6, 44(R6)	
		56	DD	000B1	PUSHAB	XPOSFAILURE	
00000000G	EF	03	FB	000B3	CLRL	-(SP)	
		50	D4	000BA	PUSHL	R6	
00000000GEF40	00	B840	D0	000BC	CALLS	#3, XPOSGET	
F2	50	03	F3	000C6	CLRL	I	0661
		FF70	31	000CA	MOVL	20(R8)[I], PAGEN[I]	0662
		04	00000000G	EF	AOBLEQ	#3, I, 6\$	
		03	13	000D4	BRW	2\$	0645
		054D	31	000D6	CMPL	MAJOR, #4	0667
		59	00000000G	EF	BEQL	8\$	
		04	B0	000E0	BRW	35\$	
34	A6	02	90	000E4	MOVZWL	MINOR, TYPE	0675
36	A6	06	90	000E8	MOV	#4, 52(R6)	0676
2C	A6	EF	9F	000EC	MOV	#2, 54(R6)	
		7E	D4	000F2	MOV	#6, 44(R6)	
		56	DD	000F4	PUSHAB	XPOSFAILURE	
00000000G	EF	03	FB	000F6	CLRL	-(SP)	
00000000G	EF	00	B8	000FD	PUSHL	R6	
		50	00000000G	EF	CALLS	#3, XPOSGET	
		24	15	0010C	MOVL	20(R8), TXTLEN	0677
		03	C0	0010E	MOVL	TXTLEN, R0	0679
		04	C6	00111	BLEQ	9\$	
		04	A5	00114	ADDL2	#3, R0	0681
34	A6	02	90	00119	DIVL2	#4, R0	
		06	90	0011D	MULW3	#4, R0, 52(R6)	
		EF	9F	00121	MOV	#2, 54(R6)	
		7E	D4	00127	MOV	#6, 44(R6)	
		56	DD	00129	PUSHAB	XPOSFAILURE	
					CLRL	-(SP)	
					PUSHL	R6	

00000000G	EF	03	FB	0012B	CALLS	#3, XPOSGET	
	01	59	D1	00132	CMPL	TYPE, #1	0683
		68	12	00135	BNEQ	11\$	
02	6E	00000000G	EF	B0	00137	MOVW	TXLEN, \$STR\$STRING
03	AE		0E	90	0013E	MOVB	#14, \$STR\$STRING+2
04	AE		01	90	00142	MOVB	#1, \$STR\$STRING+3
			68	D0	00146	MOVL	(R8), \$STR\$STRING+4
		00000000G	EF	9F	0014A	PUSHAB	STR\$FAILURE
			7E	D4	00150	CLRL	-(SP)
			5B	DD	00152	PUSHL	R11
	0C		AE	9F	00154	PUSHAB	\$STR\$STRING
			7E	D4	00157	CLRL	-(SP)
00000000G	EF	05	FB	00159	CALLS	#5, XST\$COPY	
00000000G	50	6B	3C	00160	MOVZWL	TMPSTR, R0	
00000000G	EF	50	C0	00163	ADDL2	R0, CHROUT	
00000000G	EF	6B	9E	0016A	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
00000000G	EF	07	90	00171	MOVB	#7, IOB\$+44	
		00000000G	EF	9F	00178	PUSHAB	XPOSFAILURE
			7E	D4	0017E	CLRL	-(SP)
		00000000G	EF	9F	00180	PUSHAB	IOB\$
07	00000000G	03	FB	00186	CALLS	#3, XPO\$PUT	
00000000G	EF	01	E1	0018D	BBC	#1, CMDBLK, 10\$	
00000000G	EF	00	FB	00195	CALLS	#0, SPLIT	
		FE9E	31	0019C	BRW	2\$	0683
	03	5A	E8	0019F	BLBS	FIRST_RECORD, 12\$	0694
		02CF	31	001A2	BRW	25\$	
03	00000000G	01	E0	001A5	BBS	#1, CMDBLK, 13\$	0700
		00D7	31	001AD	BRW	16\$	
		00000000G	EF	9F	001B0	PUSHAB	STR\$FAILURE
			7E	D4	001B6	CLRL	-(SP)
			5B	DD	001B8	PUSHL	R11
		00000000'	EF	9F	001BA	PUSHAB	\$STR\$STRING
			7E	D4	001C0	CLRL	-(SP)
00000000G	EF	05	FB	001C2	CALLS	#5, XST\$COPY	
00000000G	50	6B	3C	001C9	MOVZWL	TMPSTR, R0	
00000000G	EF	50	C0	001CC	ADDL2	R0, CHROUT	
00000000G	EF	6B	9E	001D3	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
00000000G	EF	07	90	001DA	MOVB	#7, IOB\$+44	
		00000000G	EF	9F	001E1	PUSHAB	XPOSFAILURE
			7E	D4	001E7	CLRL	-(SP)
		00000000G	EF	9F	001E9	PUSHAB	IOB\$
07	00000000G	03	FB	001EF	CALLS	#3, XPO\$PUT	
00000000G	EF	01	E1	001F6	BBC	#1, CMDBLK, 14\$	
00000000G	EF	00	FB	001FE	CALLS	#0, SPLIT	
		00000000G	EF	D5	00205	TSTL	HL N
			55	13	0020B	BEQL	15\$
		00000000G	EF	9F	0020D	PUSHAB	STR\$FAILURE
			7E	D4	00213	CLRL	-(SP)
			5B	DD	00215	PUSHL	R11
		00000000'	EF	9F	00217	PUSHAB	\$STR\$STRING
			7E	D4	0021D	CLRL	-(SP)
00000000G	EF	05	FB	0021F	CALLS	#5, XST\$COPY	
00000000G	50	6B	3C	00226	MOVZWL	TMPSTR, R0	
00000000G	EF	50	C0	00229	ADDL2	R0, CHROUT	
00000000G	EF	6B	9E	00230	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
00000000G	EF	07	90	00237	MOVB	#7, IOB\$+44	
		00000000G	EF	9F	0023E	PUSHAB	XPOSFAILURE

07	00000000G	EF	00000000G	7E	D4	00244	CLRL	-(SP)		
	00000000G	EF		EF	9F	00246	PUSHAB	IOB\$		
	00000000G	EF		03	FB	00248	CALLS	#3, XPO\$PUT		
				01	E1	00250	BBC	#1, CMDBLK, 15\$		
				00	FB	00252	CALLS	#0, SPLIT		
	00000000'			EF	9F	00254	PUSHAB	\$STR\$STRING2	0707	
	08			AC	DD	00256	PUSHL	TITLE		
	00000000'			EF	9F	00258	PUSHAB	\$STR\$STRING0		
00000000G		EF		03	FB	00260	CALLS	#3, XST\$JOIN		
	00000000G			EF	9F	00262	PUSHAB	STR\$FAILURE		
				7E	D4	00264	CLRL	-(SP)		
	0801			8F	BB	00266	PUSHR	#^M<R0,R11>		
				01	9F	00268	BRW	23\$		
	00000000G			EF	9F	00270	PUSHAB	STR\$FAILURE	0711	
				7E	D4	00272	CLRL	-(SP)		
				5B	DD	00274	PUSHL	R11		
	00000000'			EF	9F	00276	PUSHAB	\$STR\$STRING		
				7E	D4	00278	CLRL	-(SP)		
00000000G		EF		05	FB	00280	CALLS	#5, XST\$COPY		
		50		6B	3C	00282	MOVZWL	TMPSTR, R0		
00000000G		EF		50	C0	00284	ADDL2	R0, CHROUT		
00000000G		EF		6B	9E	00286	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
00000000G		EF		07	90	00288	MOVB	#7, IOB\$+44		
	00000000G			EF	9F	00290	PUSHAB	XPO\$FAILURE		
				7E	D4	00292	CLRL	-(SP)		
	00000000G			EF	9F	00294	PUSHAB	IOB\$		
07	00000000G	EF		03	FB	00296	CALLS	#3, XPO\$PUT		
	00000000G	EF		01	E1	00298	BBC	#1, CMDBLK, 17\$		
	00000000G	EF		00	FB	00300	CALLS	#0, SPLIT		
				EF	D5	00302	TSTL	HL N	0712	
	00000000G			12	13	00304	BEQL	18\$		
				EF	9F	00306	PUSHAB	STR\$FAILURE		
				7E	D4	00308	CLRL	-(SP)		
	00000000'			5B	DD	00310	PUSHL	R11		
				EF	9F	00312	PUSHAB	\$STR\$STRING		
	00000000G			10	11	00314	BRB	19\$		
				EF	9F	00316	PUSHAB	STR\$FAILURE	18\$:	
				7E	D4	00318	CLRL	-(SP)		
	00000000'			5B	DD	00320	PUSHL	R11		
				EF	9F	00322	PUSHAB	\$STR\$STRING	19\$:	
				7E	D4	00324	CLRL	-(SP)		
00000000G		EF		05	FB	00326	CALLS	#5, XST\$COPY		
		50		6B	3C	00328	MOVZWL	TMPSTR, R0		
00000000G		EF		50	C0	00330	ADDL2	R0, CHROUT		
00000000G		EF		6B	9E	00332	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
00000000G		EF		07	90	00334	MOVB	#7, IOB\$+44		
	00000000G			EF	9F	00336	PUSHAB	XPO\$FAILURE		
				7E	D4	00338	CLRL	-(SP)		
	00000000G			EF	9F	00340	PUSHAB	IOB\$		
07	00000000G	EF		03	FB	00342	CALLS	#3, XPO\$PUT		
	00000000G	EF		01	E1	00344	BBC	#1, CMDBLK, 20\$		
	00000000G	EF		00	FB	00346	CALLS	#0, SPLIT		
		50		EF	D0	00348	MOVL	INDENTS, R0	0714	
				70	13	00350	BEQL	21\$		
				7E	D4	00352	CLRL	-(SP)	0716	
				50	DD	00354	PUSHL	R0		
				7E	8F	00356	MOVZWL	#2307, -(SP)		

00000000G	EF	03	FB	0035D	CALLS	#3, XST\$ASCII	
		50	DD	00364	PUSHL	R0	
00000000G	EF	02	FB	00366	PUSHAB	\$STR\$STRING0	
		EF	9F	00373	CALLS	#2, XST\$JOIN	
		7E	D4	00379	PUSHAB	STR\$FAILURE	
		8F	BB	0037B	CLRL	-(SP)	
		7E	D4	0037F	PUSHR	#M<R0,R11>	
00000000G	EF	05	FB	00381	CLRL	-(SP)	
	50	6B	3C	00388	CALLS	#5, XST\$COPY	
00000000G	EF	50	C0	0038B	MOVZWL	TMPSTR, R0	
00000000G	EF	6B	9E	00392	ADDL2	R0, CHROUT	
00000000G	EF	07	90	00399	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		EF	9F	003A0	MOVAB	#7, IOB\$+44	
		7E	D4	003A6	PUSHAB	XPOS\$FAILURE	
		EF	9F	003A8	CLRL	-(SP)	
07 00000000G	EF	03	FB	003AE	PUSHAB	IOB\$	
00000000G	EF	01	E1	003B5	CALLS	#3, XPOS\$PUT	
00000000G	EF	00	FB	003BD	BBC	#1, CMDBLK, 21\$	
		EF	9F	003C4	CALLS	#0, SPLIT	
		7E	D4	003CA	PUSHAB	STR\$FAILURE	
		5B	DD	003CC	CLRL	-(SP)	
		7E	D4	003D1	PUSHL	R11	
		05	FB	003D3	PUSHL	TITLE	
00000000G	EF	6B	3C	003DA	CLRL	-(SP)	
	50	50	C0	003DD	CALLS	#5, XST\$COPY	
00000000G	EF	6B	9E	003E4	MOVZWL	TMPSTR, R0	
00000000G	EF	07	90	003EB	ADDL2	R0, CHROUT	
		EF	9F	003F2	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		7E	D4	003F8	MOVAB	#7, IOB\$+44	
		EF	9F	003FA	PUSHAB	XPOS\$FAILURE	
07 00000000G	EF	03	FB	00400	CLRL	-(SP)	
00000000G	EF	01	E1	00407	PUSHAB	IOB\$	
00000000G	EF	00	FB	0040F	CALLS	#3, XPOS\$PUT	
		EF	9F	00416	BBC	#1, CMDBLK, 22\$	
		7E	D4	0041C	CALLS	#0, SPLIT	
		5B	DD	0041E	PUSHAB	STR\$FAILURE	
		EF	9F	00420	CLRL	-(SP)	
		7E	D4	00426	PUSHL	R11	
00000000G	EF	05	FB	00428	PUSHAB	\$STR\$STRING	
	50	6B	3C	0042F	CLRL	-(SP)	
00000000G	EF	50	C0	00432	CALLS	#5, XST\$COPY	
00000000G	EF	6B	9E	00439	MOVZWL	TMPSTR, R0	
		07	90	00440	ADDL2	R0, CHROUT	
		EF	9F	00447	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		7E	D4	0044D	MOVAB	#7, IOB\$+44	
		EF	9F	0044F	PUSHAB	XPOS\$FAILURE	
07 00000000G	EF	03	FB	00455	CLRL	-(SP)	
00000000G	EF	01	E1	0045C	PUSHAB	IOB\$	
00000000G	EF	00	FB	00464	CALLS	#3, XPOS\$PUT	
00000000G	EF	01	D0	0046B	BBC	#1, CMDBLK, 24\$	
		5A	D4	00472	CALLS	#0, SPLIT	
		EF	9E	00474	MOVL	#1, HL N	
		EF	D4	0047F	CLRL	FIRST_RECORD	
		EF	D4	00485	MOVAB	LINE, LP	
		EF	D0	0048B	CLRL	INTLIN	
					CLRL	EXTLIN	
					MOVL	INDENTS, LINE_INDENT	

0718

0719

0722

0723

0729

0730

57	00000000G	EF	00000000G	EF	D0	00496	MOVL	LINE INDENT, EXTLIN	0731
	00000000G	EF	00000000G	EF	C1	004A1	ADDL3	INDENTS+4, INDENTS, R7	0732
	00000000G	EF		57	D0	004AD	MOVL	R7, WRAP	
	00000000G	EF	00000000G	EF	D0	004B4	MOVL	CMDBLK+24, RMARGIN	0733
1C	00000000G	EF		01	E1	004BF	BBC	#1, CMDBLK, 26\$	0735
FF	00000000G	EF		06	28	004C7	MOVC3	#6, P.AAP, @LP	0737
	00000000G	EF		06	C0	004D3	ADDL2	#6, LP	
	00000000G	EF		06	C0	004DA	ADDL2	#6, INTLIN	
	00000000G	EF		2F	11	004E1	BRB	27\$	0735
	00000000G	EF		57	D1	004E3	CMPL	R7, EXTLIN	0739
				26	15	004EA	BLEQ	27\$	
57	00000000G	EF	00000000G	EF	C2	004EC	SUBL2	EXTLIN, R7	
20	00000000G	6E	00000000G	FF	2C	004F3	MOVC5	#0, (SP), #32, R7, @LP	
	00000000G	EF		57	C0	004FD	ADDL2	R7, LP	
	00000000G	EF		57	C0	00504	ADDL2	R7, INTLIN	
	00000000G	EF		57	C0	0050B	ADDL2	R7, EXTLIN	
				52	D4	00512	CLRL	R2	0741
		02		59	D1	00514	CMPL	TYPE, #2	
				49	12	00517	BNEQ	30\$	
				52	D6	00519	INCL	R2	
18	00000000G	EF	00000000G	EF	9E	0051B	MOVAB	TXTPAG, PTRPAG	0748
	00000000G	EF		03	E0	00526	BBS	#3, CMDBLK, 28\$	0750
			00000000G	0A	DD	0052E	PUSHL	#10	0752
			00000000G	EF	9F	00530	PUSHAB	PTRPAG	
	00000000G	7E	00000000G	EF	3C	00536	MOVZWL	PAGEN+14, -(SP)	
		EF		03	FB	0053D	CALLS	#3, PACBAS	
			00000000G	13	11	00544	BRB	29\$	
			00000000G	EF	9F	00546	PUSHAB	PTRPAG	0754
			00000000G	EF	9F	0054C	PUSHAB	PAGEN	
	00000000G	EF		02	FB	00552	CALLS	#2, PACPAG	
	00000000G	EF		50	D0	00559	MOVL	R0, LENPAG	
			00000000G	06	11	00560	BRB	31\$	0741
			00000000G	EF	D4	00562	CLRL	LENPAG	0758
50	00000000G	EF	00000000G	EF	C3	00568	SUBL3	LENPAG, CMDBLK+24, R0	0763
	00000000G	EF	FF	A0	9E	00574	MOVAB	-1(R0), RMARGIN	
	00000000G	EF	00000000G	EF	D0	0057C	MOVL	EXTLIN, WRAP	0764
			00000000G	68	DD	00587	PUSHL	(R8)	0768
			00000000G	EF	DD	00589	PUSHL	TXLEN	
	00000000G	EF		02	FB	0058F	CALLS	#2, FMTTXT	
		09		52	E9	00596	BLBC	R2, 32\$	0770
	00000000G	EF		00	FB	00599	CALLS	#0, INSREF	0772
				1C	11	005A0	BRB	33\$	
14	00000000G	EF		01	E1	005A2	BBC	#1, CMDBLK, 33\$	0776
	00000000G	FF	40	8F	90	005AA	MOVB	#64, @LP	
			00000000G	EF	D6	005B2	INCL	LP	
			00000000G	EF	D6	005B8	INCL	INTLIN	
		6E	00000000G	EF	B0	005BE	MOVW	INTLIN, \$STR\$STRING	0782
	02	AE		0E	90	005C5	MOVB	#14, \$STR\$STRING+2	
	03	AE		01	90	005C9	MOVB	#1, \$STR\$STRING+3	
	04	AE	00000000G	EF	9E	005CD	MOVAB	LINE, \$STR\$STRING+4	
			00000000G	EF	9F	005D5	PUSHAB	STR\$FAILURE	
				7E	D4	005DB	CLRL	-(SP)	
				5B	DD	005DD	PUSHL	R11	
			0C	AE	9F	005DF	PUSHAB	\$STR\$STRING	
				7E	D4	005E2	CLRL	-(SP)	
	00000000G	EF		05	FB	005E4	CALLS	#5, XST\$COPY	

00000000G	50	6B	3C	005EB	MOVZWL	TMPSTR, R0		
00000000G	EF	50	C0	005EE	ADDL2	R0, CHROUT		
00000000G	EF	6B	9E	005F5	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
00000000G	EF	07	90	005FC	MOVB	#7, IOB\$+44		
		EF	9F	00603	PUSHAB	XPOS\$FAILURE		
	00000000G	7E	D4	00609	CLRL	-(SP)		
		EF	9F	0060B	PUSHAB	IOB\$		
03 00000000G	EF	03	FB	00611	CALLS	#3, XPOS\$PUT		
03 00000000G	EF	01	E0	00618	BBS	#1, CMDBLK, 34\$		
		0719	31	00620	BRW	73\$		
		070F	31	00623	BRW	72\$		
		50	D4	00626	CLRL	R0	0788	
OC AC 00000000G	EF	D1	00628	CMPL	MINOR, INFO			
		02	12	00630	BNEQ	36\$		
		50	D6	00632	INCL	R0		
	52	51	D2	00634	MCOML	R1, R2		
	50	52	CA	00637	BICL2	R2, R0		
	01	50	D1	0063A	CMPL	R0, #1		
		03	13	0063D	BEQL	37\$		
		00F7	31	0063F	BRW	41\$		
34 A6		OC	B0	00642	MOVW	#12, 52(R6)	0800	
36 A6		02	90	00646	MOVB	#2, 54(R6)		
2C A6		06	90	0064A	MOVB	#6, 44(R6)		
	00000000G	EF	9F	0064E	PUSHAB	XPOS\$FAILURE		
		7E	D4	00654	CLRL	-(SP)		
		56	DD	00656	PUSHL	R6		
00000000G	EF	03	FB	00658	CALLS	#3, XPOS\$GET		
00000000G	50	6B	D0	0065F	MOVL	(R8), R0	0804	
00000000G	EF	60	D0	00662	MOVL	(R0), HLLIST	0806	
00000000G	EF	04	A0	00669	MOVL	4(R0), HLDSP	0807	
	52	08	A0	00671	MOVL	8(R0), PREFIX_LEN	0808	
		25	13	00675	BEQL	38\$	0813	
		03	A2	9E	00677	MOVAB	3(R2), R0	
34 A6	50	04	C6	0067B	DIVL2	#4, R0		
	50	04	A5	0067E	MULW3	#4, R0, 52(R6)		
	50	02	90	00683	MOVB	#2, 54(R6)		
36 A6		06	90	00687	MOVB	#6, 44(R6)		
2C A6		EF	9F	0068B	PUSHAB	XPOS\$FAILURE		
	00000000G	7E	D4	00691	CLRL	-(SP)		
		56	DD	00693	PUSHL	R6		
00000000G	EF	03	FB	00695	CALLS	#3, XPOS\$GET		
	6E	52	B0	0069C	MOVW	PREFIX_LEN, \$STR\$STRING	0815	
	AE	0E	90	0069F	MOVB	#14, \$STR\$STRING+2		
	02	01	90	006A3	MOVB	#1, \$STR\$STRING+3		
	03	6B	D0	006A7	MOVL	(R8), \$STR\$STRING+4		
	AE	EF	9F	006AB	PUSHAB	\$STR\$FAILURE		
		7E	D4	006B1	CLRL	-(SP)		
	00000000G	EF	9F	006B3	PUSHAB	\$STR\$TARGET		
		AE	9F	006B9	PUSHAB	\$STR\$STRING		
	OC	7E	D4	006BC	CLRL	-(SP)		
00000000G	EF	05	FB	006BE	CALLS	#5, XST\$COPY	0819	
	34 A6	04	B0	006C5	MOVW	#4, 52(R6)		
	36 A6	02	90	006C9	MOVB	#2, 54(R6)		
	2C A6	06	90	006CD	MOVB	#6, 44(R6)		
	00000000G	EF	9F	006D1	PUSHAB	XPOS\$FAILURE		
		7E	D4	006D7	CLRL	-(SP)		
		56	DD	006D9	PUSHL	R6		

00000000G	EF	00	03	FB	006DB	CALLS	#3, XPOSGET	0820
	52		B8	D0	006E2	MOVL	@0(R8), SUFFIX_LEN	0824
		03	25	13	006E6	BEQL	39\$	
34	A6		A2	9E	006E8	MOVAB	3(R2), R0	
	50		04	C6	006EC	DIVL2	#4, R0	
	50		04	A5	006EF	MULW3	#4, R0, 52(R6)	
36	A6		02	90	006F4	MOVB	#2, 54(R6)	
2C	A6		06	90	006F8	MOVB	#6, 44(R6)	
		00000000G	EF	9F	006FC	PUSHAB	XPOSFAILURE	
			7E	D4	00702	CLRL	-(SP)	
			56	DD	00704	PUSHL	R6	
00000000G	EF		03	FB	00706	CALLS	#3, XPOSGET	0826
	6E		52	B0	0070D	MOVL	SUFFIX_LEN, \$STR\$STRING	
02	AE		0E	90	00710	MOVB	#14, \$STR\$STRING+2	
03	AE		01	90	00714	MOVB	#1, \$STR\$STRING+3	
04	AE		68	D0	00718	MOVL	(R8), \$STR\$STRING+4	
		00000000G	EF	9F	0071C	PUSHAB	\$STR\$FAILURE	
			7E	D4	00722	CLRL	-(SP)	
		00000000G	EF	9F	00724	PUSHAB	\$STR\$TARGET	
		0C	AE	9F	0072A	PUSHAB	\$STR\$STRING	
			7E	D4	0072D	CLRL	-(SP)	
00000000G	EF		05	FB	0072F	CALLS	#5, XST\$COPY	0645
			F904	31	00736	BRW	2\$	0829
			52	D4	00739	CLRL	R2	
10	AC	00000000G	EF	D1	0073B	CMPL	MINOR, TXT	
			02	12	00743	BNEQ	42\$	
			52	D6	00745	INCL	R2	
			50	D4	00747	CLRL	R0	
14	AC	00000000G	EF	D1	00749	CMPL	MINOR, TXT_U	
			02	12	00751	BNEQ	43\$	
			50	D6	00753	INCL	R0	
	50		52	C8	00755	BISL2	R2, R0	
	53		51	D2	00758	MCOML	R1, R3	
	50		53	CA	0075B	BICL2	R3, R0	
	01		50	D1	0075E	CMPL	R0, #1	
			D3	12	00761	BNEQ	40\$	0840
34	A6		04	B0	00763	MOVL	#4, 52(R6)	
36	A6		02	90	00767	MOVB	#2, 54(R6)	
2C	A6		06	90	0076B	MOVB	#6, 44(R6)	
		00000000G	EF	9F	0076F	PUSHAB	XPOSFAILURE	
			7E	D4	00775	CLRL	-(SP)	
			56	DD	00777	PUSHL	R6	
00000000G	EF		03	FB	00779	CALLS	#3, XPOSGET	0841
00000000G	EF	00	B8	D0	00780	MOVL	@0(R8), TXTLEN	0846
	50	00000000G	EF	D0	00788	MOVL	TXTLEN, R0	
			24	13	0078F	BEQL	44\$	
	50		03	C0	00791	ADDL2	#3, R0	0848
	50		04	C6	00794	DIVL2	#4, R0	
34	A6		04	A5	00797	MULW3	#4, R0, 52(R6)	
	50		02	90	0079C	MOVB	#2, 54(R6)	
36	A6		06	90	007A0	MOVB	#6, 44(R6)	
2C	A6		EF	9F	007A4	PUSHAB	XPOSFAILURE	
		00000000G	7E	D4	007AA	CLRL	-(SP)	
			56	DD	007AC	PUSHL	R6	
00000000G	EF		03	FB	007AE	CALLS	#3, XPOSGET	
00000000G	EF	00000000G	EF	9E	007B5	MOVAB	LINE, LP	0853
		00000000G	EF	D4	007C0	CLRL	INTLIN	

00000000G	EF	00000000G	EF	D4	007C6	CLRL	EXTLIN	0854
00000000G	EF	00000000G	EF	D0	007CC	MOVL	INDENTS, LINE INDENT	0855
00000000G	EF	00000000G	EF	D0	007D7	MOVL	LINE INDENT, EXTLIN	0856
00000000G	EF	00000000G	EF	C1	007E2	ADDL3	INDENTS+4, INDENTS, WRAP	0857
00000000G	EF	00000000G	EF	D0	007F2	MOVL	CMDBLK+24, RMARGIN	0859
03	03	00000000G	EF	5A	007FD	BLBS	FIRST_RECORD, 45\$	0865
			02CF	31	00800	BRW	58\$	0866
			01	E0	00803	BBS	#1, CMDBLK, 46\$	0868
			00D7	31	0080B	BRW	49\$	
		00000000G	EF	9F	0080E	PUSHAB	STR\$FAILURE	
			7E	D4	00814	CLRL	-(SP)	
		00000000'	5B	DD	00816	PUSHL	R11	
			EF	9F	00818	PUSHAB	\$STR\$STRING	
			7E	D4	0081E	CLRL	-(SP)	
		00000000G	EF	05	FB	CALLS	#5, XST\$COPY	
			50	3C	00827	MOVZWL	TMPSTR, R0	
		00000000G	EF	50	C0	ADDL2	R0, CHROUT	
		00000000G	EF	6B	9E	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		00000000G	EF	07	90	MOVB	#7, IOB\$+44	
			EF	9F	0083F	PUSHAB	XPO\$FAILURE	
		00000000G	EF	7E	D4	CLRL	-(SP)	
			EF	9F	00847	PUSHAB	IOB\$	
		00000000G	EF	03	FB	CALLS	#3, XPO\$PUT	
		00000000G	EF	01	E1	BBC	#1, CMDBLK, 47\$	
		00000000G	EF	00	FB	CALLS	#0, SPLIT	
			EF	D5	00863	TSTL	HL N	0870
			55	13	00869	BEQL	48\$	
		00000000G	EF	9F	0086B	PUSHAB	STR\$FAILURE	
			7E	D4	00871	CLRL	-(SP)	
			5B	DD	00873	PUSHL	R11	
		00000000'	EF	9F	00875	PUSHAB	\$STR\$STRING	
			7E	D4	0087B	CLRL	-(SP)	
		00000000G	EF	05	FB	CALLS	#5, XST\$COPY	
			50	3C	00884	MOVZWL	TMPSTR, R0	
		00000000G	EF	50	C0	ADDL2	R0, CHROUT	
		00000000G	EF	6B	9E	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		00000000G	EF	07	90	MOVB	#7, IOB\$+44	
			EF	9F	0089C	PUSHAB	XPO\$FAILURE	
		00000000G	EF	7E	D4	CLRL	-(SP)	
			EF	9F	008A4	PUSHAB	IOB\$	
		00000000G	EF	03	FB	CALLS	#3, XPO\$PUT	
		00000000G	EF	01	E1	BBC	#1, CMDBLK, 48\$	
		00000000G	EF	00	FB	CALLS	#0, SPLIT	
		00000000'	EF	9F	008C0	PUSHAB	\$STR\$STRING2	0872
			08	AC	DD	PUSHL	TITLE	
		00000000'	EF	9F	008C9	PUSHAB	\$STR\$STRINGO	
		00000000G	EF	03	FB	CALLS	#3, XST\$JOIN	
			EF	9F	008D6	PUSHAB	STR\$FAILURE	
			7E	D4	008DC	CLRL	-(SP)	
		0801	8F	BB	008DE	PUSHR	#*M<R0,R11>	
			01	9F	31	BRW	56\$	
		00000000G	EF	9F	008E5	PUSHAB	STR\$FAILURE	0876
			7E	D4	008EB	CLRL	-(SP)	
			5B	DD	008ED	PUSHL	R11	
		00000000'	EF	9F	008EF	PUSHAB	\$STR\$STRING	
			7E	D4	008F5	CLRL	-(SP)	
		00000000G	EF	05	FB	CALLS	#5, XST\$COPY	

00000000G	50	68	3C	008FE	MOVZWL	TMPSTR, R0
00000000G	EF	50	C0	00901	ADDL2	R0, CHROUT
00000000G	EF	68	9E	00908	MOVAB	\$IOB\$OUTPUT, IOB\$+68
		07	90	0090F	MOVB	#7, IOB\$+44
	00000000G	EF	9F	00916	PUSHAB	XPOSFAILURE
		7E	D4	0091C	CLRL	-(SP)
	00000000G	EF	9F	0091E	PUSHAB	IOB\$
07 00000000G	EF	03	FB	00924	CALLS	#3, XPO\$PUT
00000000G	EF	01	E1	0092B	BBC	#1, CMDBLK, 50\$
00000000G	EF	00	FB	00933	CALLS	#0, SPLIT
	00000000G	EF	D5	0093A	TSTL	HL N
		12	13	00940	BEQL	51\$
	00000000G	EF	9F	00942	PUSHAB	STR\$FAILURE
		7E	D4	00948	CLRL	-(SP)
	00000000'	5B	DD	0094A	PUSHL	R11
		EF	9F	0094C	PUSHAB	\$STR\$STRING
	00000000G	10	11	00952	BRB	52\$
		EF	9F	00954	PUSHAB	STR\$FAILURE
		7E	D4	0095A	CLRL	-(SP)
	00000000'	5B	DD	0095C	PUSHL	R11
		EF	9F	0095E	PUSHAB	\$STR\$STRING
		7E	D4	00964	CLRL	-(SP)
00000000G	EF	05	FB	00966	CALLS	#5, XST\$COPY
00000000G	50	68	3C	0096D	MOVZWL	TMPSTR, R0
00000000G	EF	50	C0	00970	ADDL2	R0, CHROUT
00000000G	EF	68	9E	00977	MOVAB	\$IOB\$OUTPUT, IOB\$+68
		07	90	0097E	MOVB	#7, IOB\$+44
	00000000G	EF	9F	00985	PUSHAB	XPOSFAILURE
		7E	D4	0098B	CLRL	-(SP)
	00000000G	EF	9F	0098D	PUSHAB	IOB\$
07 00000000G	EF	03	FB	00993	CALLS	#3, XPO\$PUT
00000000G	EF	01	E1	0099A	BBC	#1, CMDBLK, 53\$
00000000G	EF	00	FB	009A2	CALLS	#0, SPLIT
	50 00000000G	EF	D0	009A9	MOVL	INDENTS, R0
		70	13	009B0	BEQL	54\$
		7E	D4	009B2	CLRL	-(SP)
		50	DD	009B4	PUSHL	R0
00000000G	7E 0903	8F	3C	009B6	MOVZWL	#2307, -(SP)
	EF	03	FB	009BB	CALLS	#3, XST\$ASCII
		50	DD	009C2	PUSHL	R0
00000000G	00000000'	EF	9F	009C4	PUSHAB	\$STR\$STRINGO
		02	FB	009CA	CALLS	#2, XST\$JOIN
	00000000G	EF	9F	009D1	PUSHAB	STR\$FAILURE
		7E	D4	009D7	CLRL	-(SP)
	0801	8F	BB	009D9	PUSHR	#*M<R0,R11>
		7E	D4	009DD	CLRL	-(SP)
00000000G	EF	05	FB	009DF	CALLS	#5, XST\$COPY
00000000G	50	68	3C	009E6	MOVZWL	TMPSTR, R0
00000000G	EF	50	C0	009E9	ADDL2	R0, CHROUT
00000000G	EF	68	9E	009F0	MOVAB	\$IOB\$OUTPUT, IOB\$+68
00000000G	EF	07	90	009F7	MOVB	#7, IOB\$+44
	00000000G	EF	9F	009FE	PUSHAB	XPOSFAILURE
		7E	D4	00A04	CLRL	-(SP)
	00000000G	EF	9F	00A06	PUSHAB	IOB\$
07 00000000G	EF	03	FB	00A0C	CALLS	#3, XPO\$PUT
00000000G	EF	01	E1	00A13	BBC	#1, CMDBLK, 54\$
00000000G	EF	00	FB	00A1B	CALLS	#0, SPLIT

0877

0879

0881

		00000000G	EF	9F	00A22	54\$:	PUSHAB	STR\$FAILURE	0883
			7E	D4	00A28		CLRL	-(SP)	
			5B	DD	00A2A		PUSHL	R11	
		08	AC	DD	00A2C		PUSHL	TITLE	
			7E	D4	00A2F		CLRL	-(SP)	
	00000000G	EF	05	FB	00A31		CALLS	#5, XST\$COPY	
		50	6B	3C	00A38		MOVZWL	TMPSTR, R0	
	00000000G	EF	50	C0	00A3B		ADDL2	R0, CHROUT	
	00000000G	EF	6B	9E	00A42		MOVAB	\$IOB\$OUTPUT, IOB\$+68	
	00000000G	EF	07	90	00A49		MOVB	#7, IOB\$+44	
		00000000G	EF	9F	00A50		PUSHAB	XPOS\$FAILURE	
			7E	D4	00A56		CLRL	-(SP)	
		00000000G	EF	9F	00A58		PUSHAB	IOB\$	
	00000000G	EF	03	FB	00A5E		CALLS	#3, XPO\$PUT	
07	00000000G	EF	01	E1	00A65		BBC	#1, CMDBLK, 55\$	
	00000000G	EF	00	FB	00A6D		CALLS	#0, SPLIT	
		00000000G	EF	9F	00A74	55\$:	PUSHAB	STR\$FAILURE	0884
			7E	D4	00A7A		CLRL	-(SP)	
			5B	DD	00A7C		PUSHL	R11	
		00000000'	EF	9F	00A7E		PUSHAB	\$STR\$STRING	
			7E	D4	00A84	56\$:	CLRL	-(SP)	
	00000000G	EF	05	FB	00A86		CALLS	#5, XST\$COPY	
		50	6B	3C	00A8D		MOVZWL	TMPSTR, R0	
	00000000G	EF	50	C0	00A90		ADDL2	R0, CHROUT	
	00000000G	EF	6B	9E	00A97		MOVAB	\$IOB\$OUTPUT, IOB\$+68	
	00000000G	EF	07	90	00A9E		MOVB	#7, IOB\$+44	
		00000000G	EF	9F	00AA5		PUSHAB	XPOS\$FAILURE	
			7E	D4	00AAB		CLRL	-(SP)	
		00000000G	EF	9F	00AAD		PUSHAB	IOB\$	
	00000000G	EF	03	FB	00AB3		CALLS	#3, XPO\$PUT	
07	00000000G	EF	01	E1	00ABA		BBC	#1, CMDBLK, 57\$	
	00000000G	EF	00	FB	00AC2		CALLS	#0, SPLIT	
	00000000G	EF	01	D0	00AC9	57\$:	MOVL	#1, HL_N	0887
			5A	D4	00AD0		CLRL	FIRST RECORD	0888
	1A 00000000G	EF	01	E1	00AD2	58\$:	BBC	#1, CMDBLK, 59\$	0891
	FF 00000000'	EF	05	28	00ADA		MOVCL3	#5, P.ABF, @LP	
	00000000G	EF	05	C0	00AE6		ADDL2	#5, LP	
	00000000G	EF	05	C0	00AED		ADDL2	#5, INTLIN	
03	00000000G	EF	04	E0	00AF4	59\$:	BBS	#4, CMDBLK, 61\$	0893
		01	01	31	00AFC	60\$:	BRW	66\$	
	00000000G	EF	00	D1	00AFF	61\$:	CMPL	CMDBLK+20, HL_N	0895
			17	18	00B0A		BGEQ	62\$	
	FFFFFFFF	8F	00	D1	00B0C		CMPL	CMDBLK+20, #-1	0897
			E3	12	00B17		BNEQ	60\$	
	14	AC	00	D1	00B19		CMPL	MINOR, TXT_U	0898
			D9	13	00B21		BEQL	60\$	
	00000000G	EF	00	9E	00B23	62\$:	MOVAB	TXTSCT, PTRSCT	0909
			EF	D4	00B2E		CLRL	LENSCT	0910
		0F	00	93	00B34		BITB	PAGEN, #15	0912
			2F	13	00B3B		BEQL	63\$	
		00000000G	EF	9F	00B3D		PUSHAB	PTRSCT	0918
		00000000G	EF	9F	00B43		PUSHAB	PAGEN	
	00000000G	EF	02	FB	00B49		CALLS	#2, PACSEC	
		50	00	C0	00B50		ADDL2	LENSCT, R0	
	00000000G	EF	01	A0	9E	00B57	MOVAB	1(R0), LENSCT	
	00000000G	FF	2D	90	00B5F		MOVB	#45, @PTRSCT	0919
		00000000G	EF	D6	00B66		INCL	PTRSCT	

		00000000G	EF	DD	00B6C	63\$:	PUSHL	HLDSP		0922
		00000000G	EF	9F	00B72		PUSHAB	PTRSCT		
		00000000G	EF	DD	00B78		PUSHL	HLLIST		
	00000000G	EF	03	FB	00B7E		CALLS	#3, PACXXX		
	00000000G	EF	50	CO	00B85		ADDL2	R0, LENSCT		
		50	00000000G	EF	3C	00B8C	MOVZWL	PRESTR, R0		0926
			OF	13	00B93		BEQL	64\$		
		00000000G	EF	DD	00B95		PUSHL	PRESTR+4		0928
			50	DD	00B9B		PUSHL	R0		
	00000000G	EF	02	FB	00B9D		CALLS	#2, FMTTXT		
		00000000G	EF	9F	00BA4	64\$:	PUSHAB	TXTSCT		0933
		00000000G	EF	DD	00BAA		PUSHL	LENSCT		
	00000000G	EF	02	FB	00BB0		CALLS	#2, FMTTXT		
		54	00000000G	EF	3C	00BB7	MOVZWL	SFXSTR, R4		0937
			1F	13	00BBE		BEQL	65\$		
54	00	00000000'	EF	01	2D	00BC0	CMPC5	#1, P.ABG, #0, R4, @SFXSTR+4		0941
		00000000G	FF		00BC9					
		00000000G	OF	13	00BCE		BEQL	65\$		
			EF	DD	00BD0		PUSHL	SFXSTR+4		0944
			54	DD	00BD6		PUSHL	R4		
	00000000G	EF	02	FB	00BD8		CALLS	#2, FMTTXT		
21	00000000G	EF	01	E0	00BDF	65\$:	BBS	#1, CMDBLK, 67\$		0948
	00000000G	FF	20	90	00BE7		MOVB	#32, @LP		
		00000000G	EF	D6	00BEE		INCL	LP		
		00000000G	EF	D6	00BF4		INCL	INTLIN		
		00000000G	EF	D6	00BFA		INCL	EXTLIN		
16	00000000G	EF	01	E1	00C00	66\$:	BBC	#1, CMDBLK, 68\$		0951
	00000000G	FF	8F	90	00C08	67\$:	MOVB	#64, @LP		0953
		00000000G	EF	D6	00C10		INCL	LP		
		00000000G	EF	D6	00C16		INCL	INTLIN		
			3B	11	00C1C		BRB	69\$		0951
57	00000000G	EF	00000000G	EF	C1	00C1E	68\$:	ADDL3	INDENTS+4, INDENTS, R7	0955
	00000000G	EF	57	D1	00C2A		CMPL	R7, EXTLIN		
			26	15	00C31		BLEQ	69\$		
		57	00000000G	EF	C2	00C33	SUBL2	EXTLIN, R7		
57	20	6E	00000000G	00	2C	00C3A	MOVCS	#0, (SP), #32, R7, @LP		
			00000000G	FF		00C3F				
	00000000G	EF	57	CO	00C44		ADDL2	R7, LP		
	00000000G	EF	57	CO	00C4B		ADDL2	R7, INTLIN		
	00000000G	EF	57	CO	00C52		ADDL2	R7, EXTLIN		
	00000000G	EF	00000000G	EF	9E	00C59	69\$:	MOVAB	TXTPAG, PTRPAG	0961
18	00000000G	EF	03	E0	00C64		BBS	#3, CMDBLK, 70\$		0963
			0A	DD	00C6C		PUSHL	#10		0969
		00000000G	EF	9F	00C6E		PUSHAB	PTRPAG		
		00000000G	EF	3C	00C74		MOVZWL	PAGEN+14, -(SP)		
	00000000G	EF	03	FB	00C7B		CALLS	#3, PACBAS		
			13	11	00C82		BRB	71\$		
		00000000G	EF	9F	00C84	70\$:	PUSHAB	PTRPAG		0976
		00000000G	EF	9F	00C8A		PUSHAB	PAGEN		
	00000000G	EF	02	FB	00C90		CALLS	#2, PACPAG		
	00000000G	EF	50	DO	00C97	71\$:	MOVL	R0, LENPAG		
50	00000000G	EF	00000000G	EF	C3	00C9E	SUBL3	LENPAG, CMDBLK+24, R0		0982
	00000000G	EF	FF	A0	9E	00CAA	MOVAB	-1(R0), RMARGIN		
	00000000G	EF	00000000G	EF	DO	00CB2	MOVL	EXTLIN, WRAP		0983
				68	DD	00CBD	PUSHL	(R8)		0987
		00000000G	EF	DD	00CBF		PUSHL	TXLEN		
	00000000G	EF	02	FB	00CC5		CALLS	#2, FMTTXT		

CAPTION
V04-000

CAPTION - Generate figure, table and example ca
HDR_X - Process .HEADER x captions

H 11
15-Sep-1984 23:50:23
14-Sep-1984 13:05:37

VAX-11 B11sg-32 V4.0-742
[RUNOFF.SRC]CAPTION.BLI;1

Page 27
(2)

CAP
V04

```
00000000G EF 00 FB 00CCC CALLS #0, INSREF
02 6E 00000000G EF 80 00CD3 MOVW INTLIN, $STR$STRING
03 AE 00000000G 0E 90 00CDA MOVW #14, $STR$STRING+2
04 AE 00000000G 01 90 00CDE MOVW #1, $STR$STRING+3
00000000G EF 9E 00CE2 MOVAB LINE, $STR$STRING+4
00000000G EF 9F 00CEA PUSHAB STR$FAILURE
7E D4 00CF0 CLRL -(SP)
5B DD 00CF2 PUSHL R11
OC AE 9F 00CF4 PUSHAB $STR$STRING
7E D4 00CF7 CLRL -(SP)
00000000G EF 05 FB 00CF9 CALLS #5, XST$COPY
50 3C 00D00 MOVZWL TMPSTR, R0
00000000G EF 50 C0 00D03 ADDL2 R0, CHROUT
00000000G EF 6B 9E 00D0A MOVAB $IOB$OUTPUT, IOB$+68
00000000G EF 07 90 00D11 MOVW #7, IOB$+44
00000000G EF 9F 00D18 PUSHAB XPOS$FAILURE
7E D4 00D1E CLRL -(SP)
00000000G EF 9F 00D20 PUSHAB IOB$
07 00000000G EF 03 FB 00D26 CALLS #3, XPOS$PUT
00000000G EF 01 E1 00D2D BBC #1, CMDBLK, 73$
00000000G EF 00 FB 00D35 72$: CALLS #0, SPLIT
00000000G EF 9E 00D3C 73$: MOVAB LINE, LP
00000000G EF D4 00D47 CLRL INTLIN
F2E7 EF D4 00D4D CLRL EXTLIN
2E A6 10 88 00D53 BRW 2$
2C A6 02 90 00D5A 74$: BISB2 #16, 46(R6)
00000000G EF 9F 00D5E MOVW #2, 44(R6)
7E D4 00D64 PUSHAB XPOS$FAILURE
56 DD 00D66 CLRL -(SP)
00000000G EF 03 FB 00D68 PUSHL R6
04 00D6F CALLS #3, XPOS$CLOSE
RET
```

0991
0995
0996
0634
1003
1004

; Routine Size: 3440 bytes, Routine Base: \$CODE\$ + 0000


```
691 1005 1 %SBTTL 'STCHDR - Process the .SEND TOC n captions temp file'
692 1006 1 GLOBAL ROUTINE stchr : NOVALUE =
693 1007 1 ++
694 1008 1
695 1009 1 FUNCTIONAL DESCRIPTION:
696 1010 1
697 1011 1     This routine reads the temp file, which contains page references
698 1012 1     and .SEND TOC records.
699 1013 1
700 1014 1     The .SEND TOC records are formatted into captions.
701 1015 1
702 1016 1 FORMAL PARAMETERS:
703 1017 1
704 1018 1     None
705 1019 1
706 1020 1 IMPLICIT INPUTS:
707 1021 1
708 1022 1     nxtstc - next .SEND TOC 'n' value to process
709 1023 1     cmdblk - command line information block
710 1024 1
711 1025 1 IMPLICIT OUTPUTS:
712 1026 1
713 1027 1     None
714 1028 1
715 1029 1 ROUTINE VALUE:
716 1030 1 COMPLETION CODES:
717 1031 1
718 1032 1     None
719 1033 1
720 1034 1 SIDE EFFECTS:
721 1035 1
722 1036 1     None
723 1037 1
724 1038 1 --
725 1039 1
726 1040 1 BEGIN
727 1041 1
728 1042 1     Walk through the file until there are no more .SEND TOC 'n'
729 1043 1     groups left to process only if we are processing HL 1's
730 1044 1
731 1045 1 WHILE (.nxtstc NEQ infinity) AND (.cmdblk [contents$g_headers] NEQ 0) DO
732 1046 1     BEGIN
733 1047 1
734 1048 1     LOCAL
735 1049 1         stcn,
736 1050 1         first_record;
737 1051 1
738 1052 1     first_record = true;
739 1053 1     stcn = .nxtstc;
740 1054 1     nxtstc = infinity;
741 1055 1     $xpo_open (iob = stciob, options = input, attributes = BINARY
742 1056 1     $IF %BLISS (BLISS32) %THEN , failure = open_error %FI
743 1057 1     );
744 1058 1
745 1059 1     WHILE $xpo_get (iob = stciob, fullwords = 2) EQL xpo$_normal DO
746 1060 1         BEGIN
747 1061 1         !
! Value we're processing this time
! Value to process next time
```

```
748 1062 4      ! Process a record group
749 1063 4      !
750 1064 4      BIND
751 1065 4      record_header = .stciob [iob$a_data] : VECTOR;
752 1066 4
753 1067 4      major = .record_header [0];
754 1068 4      minor = .record_header [1];
755 1069 4
756 1070 4      SELECTONE true OF
757 1071 4      SET
758 1072 4
759 1073 4      [(.major EQL maj_runoff) AND (.minor EQL min_page)] :
760 1074 3      BEGIN
761 1075 3      ! Process a page reference
762 1076 3
763 1077 3      $xpo_get (iob = stciob, fullwords = page_sct_size);
764 1078 3      BEGIN
765 1079 6
766 1080 6      BIND
767 1081 6      page_ref = .stciob [iob$a_data] : VECTOR;
768 1082 6
769 1083 6      MAP
770 1084 6      pagen : VECTOR;
771 1085 6
772 1086 6      INCR i FROM 0 TO page_sct_size - 1 DO
773 1087 6      pagen [i] = .page_ref [i];
774 1088 6
775 1089 6      END;
776 1090 3      END;
777 1091 4
778 1092 4      [(.major EQL maj_send) AND (ABS (.minor) NEQ .stcn)] :
779 1093 4      BEGIN
780 1094 3
781 1095 3      ! A .SEND TOC with a value of n which does not match the
782 1096 3      ! value we're processing. Remember it if it is the smallest
783 1097 3      ! one which is larger than the value we're processing.
784 1098 3
785 1099 3      IF ABS (.minor) GTR .stcn THEN nxtstc = MIN (ABS (.minor), .nxtstc);
786 1100 3
787 1101 3
788 1102 3      ! Now skip over the text if any
789 1103 3
790 1104 3      $xpo_get (iob = stciob, fullwords = 1);
791 1105 3
792 1106 3      IF (.stciob [iob$a_data]) NEQ 0
793 1107 3      THEN
794 1108 3      P $xpo_get (iob = stciob,
795 1109 3      fullwords = CH$ALLOCATION (.(.stciob [iob$a_data])));
796 1110 3
797 1111 3
798 1112 3      END;
799 1113 4
800 1114 4      [(.major EQL maj_send) AND (.minor EQL .stcn)] :
801 1115 3      BEGIN
802 1116 3
803 1117 3      ! A caption title
804 1118 3
```

```
805 1119 5 LOCAL
806 1120 ptr,
807 1121 len;
808 1122
809 1123 clr_line ();
810 1124 line_indent = .indents [0];
811 1125 extln = .line_indent;
812 1126 wrap = .indents [0] + .indents [1];
813 1127 rmargin = .cndblk [contents$g_page_width];
814 1128 major = maj_runoff;
815 1129 ! Make .SEND TOC +n look like a caption.
816 1130
817 1131 IF .first_record AND .stcn LEQ 3
818 1132 THEN
819 1133 BEGIN
820 1134 First one seen - generate title
821 1135
822 1136 IF .cndblk [contents$v_tms11]
823 1137 THEN
824 1138 BEGIN
825 1139 BIND
826 1140 tms_titles = UPLIT (0, tms_ex, tms_fg, tms_tb) : VECTOR;
827 1141 put ('');
828 1142 IF .hl_n NEQ 0 THEN put ('[va15]');
829 1143 put ($str_concat ('*c0n*', .tms_titles [.stcn], '@@'));
830 1144 END
831 1145 ELSE
832 1146 BEGIN
833 1147 BIND
834 1148 rno_titles = UPLIT (0, rno_ex, rno_fg, rno_tb) : VECTOR,
835 1149 title = .rno_titles [.stcn] : $str_descriptor ();
836 1150
837 1151 put ('.TEST PAGE 7');
838 1152 IF .hl_n NEQ 0 THEN put ('.SK2') ELSE put ('.SK1');
839 1153
840 1154 IF .indents [0] NEQ 0
841 1155 THEN
842 1156 put ($str_concat ('.I-', $str_ascii (.indents [0])));
843 1157
844 1158 put (title);
845 1159 put ('.BLANK');
846 1160 END;
847 1161
848 1162 first_record = false;
849 1163 END;
850 1164
851 1165 hl_n = 1;
852 1166
853 1167 IF .cndblk [contents$v_tms11] THEN literal_text ('*c1n*');
854 1168
855 1169 !
856 1170 ! Process the record
857 1171
858 1172
859 1173
860 1174
861 1175 5
```



```
1176 5  
1177 5  
1178 5  
1179 5  
1180 5  
1181 5  
1182 5  
1183 5  
1184 6  
1185 6  
1186 6  
1187 6  
1188 6  
1189 6  
1190 7  
1191 7  
1192 7  
1193 7  
1194 7  
1195 7  
1196 7  
1197 7  
1198 7  
1199 7  
1200 7  
1201 7  
1202 7  
1203 7  
1204 8  
1205 8  
1206 8  
1207 8  
1208 8  
1209 8  
1210 9  
1211 9  
1212 9  
1213 9  
1214 9  
1215 9  
1216 9  
1217 9  
1218 8  
1219 8  
1220 8  
1221 8  
1222 8  
1223 8  
1224 8  
1225 8  
1226 7  
1227 7  
1228 6  
1229 6  
1230 5  
1231 5  
1232 5
```

```
!  
$xpo_get (iob = stciob, fullwords = 1);  
len = (.stciob [iob$a_data]);  
$xpo_get (iob = stciob, fullwords = CH$ALLOCATION (.len));  
ptr = CH$FIND_CH (.len, CH$PTR (.stciob [iob$a_data]), '%i,');  
  
IF .cmdblk [contents$v_include_sections]  
THEN  
  BEGIN  
    : Generate a section number if the user requested one.  
    IF NOT CH$FAIL (.ptr)  
    THEN  
      BEGIN  
        : User supplied a section number  
        LOCAL  
          sec_l,  
          sec_p;  
  
        sec_p = CH$PTR (.stciob [iob$a_data]);  
        sec_l = CH$DIFF (.ptr, .sec_p);  
        strip (sec_l, sec_p);  
  
        IF .sec_l NEQ 0  
        THEN  
          BEGIN  
            : User specified a non-null section number  
            IF .pagen [sct_typ] NEQ 0  
            THEN  
              BEGIN  
                : Document is divided into sections  
                ptrsct = CH$PTR (txtsct);  
                lensct = pacsec (pagen, ptrsct);  
                fmttxt (.lensct, CH$PTR (txtsct));  
                fmttxt (1, CH$PTR (UPLIT ('-')));  
                END;  
  
                fmttxt (.sec_l, .sec_p);  
  
                IF NOT .cmdblk [contents$v_tms11]  
                THEN  
                  write_char (%' ', counts_visually);  
  
                END;  
  
            END;  
  
          END;  
  
        END;  
  
      IF .cmdblk [contents$v_tms11]
```

```
919      1233      5      THEN
920      1234      6      write_char (XC'@')
921      1235      6      ELSE
922      1236      6      pad ((.indents [0] + .indents [1] - .extlin));
923      1237      6
924      1238      6      |
925      1239      6      | Generate page number to get its length.
926      1240      6      |
927      1241      6      ptrpag = CH$PTR (txtpag);
928      1242      6
929      1243      6      IF NOT .cmdblk [contents$v_standard_page]
930      1244      6      THEN
931      1245      6      BEGIN
932      1246      6      |
933      1247      6      | Use running page number
934      1248      6      |
935      1249      6      | lenpag = pacbas (.pagen [sct_run_page], ptrpag, 10);
936      1250      6      |
937      1251      6      | END
938      1252      6      ELSE
939      1253      6      BEGIN
940      1254      6      |
941      1255      6      | Supply the usual page number
942      1256      6      |
943      1257      6      | lenpag = pacpag (pagen, ptrpag);
944      1258      6      |
945      1259      6      | END;
946      1260      6      |
947      1261      6      | rmargin = .cmdblk [contents$g_page_width] - .lenpag - 1;
948      1262      6      | wrap = .extlin;
949      1263      6      |
950      1264      6      | Process the text
951      1265      6      |
952      1266      6      | IF CH$FAIL (.ptr)
953      1267      6      | THEN
954      1268      6      | ptr = CH$PTR (.stciob [iob$a_data])
955      1269      6      | ELSE
956      1270      6      | BEGIN
957      1271      6      | |
958      1272      6      | | There was a section number in the line. Point past
959      1273      6      | | the comma and update the length.
960      1274      6      | |
961      1275      6      | | ptr = CH$PLUS (.ptr, 1);
962      1276      6      | | len = .len - CH$DIFF (.ptr, CH$PTR (.stciob [iob$a_data]));
963      1277      6      | | END;
964      1278      6      | strip (len, ptr);
965      1279      6      | fmttxt (.len, .ptr);
966      1280      6      |
967      1281      6      | Insert page reference into page if desired.
968      1282      6      |
969      1283      6      | insref ();
970      1284      6      |
971      1285      6      | Write line to output file
972      1286      6      |
973      1287      6      | put ((.intlin, CH$PTR (line)));
974      1288      6      | clr_line ();
975      1289      6      | END;
```

```
976 1290 4      [(.major EQL maj_send) AND (.minor EQL -.stcn)] :
977 1291 3      BEGIN
978 1292 3      |
979 1293 3      | Insert directly into .RNT file
980 1294 3      |
981 1295 3      | LOCAL
982 1296 3      |   len;
983 1297 3      |
984 1298 3      |   $xpo_get (iob = stciob, fullwords = 1);
985 1299 3      |   len = .(.stciob [iob$a_data]);
986 1300 3      |   $xpo_get (iob = stciob, fullwords = CH$ALLOCATION (.len));
987 1301 3      |   insrnt (.len, CH$PTR (.stciob [iob$a_data]));
988 1302 4      |   END;
989 1303 4      |
990 1304 4      |   TES;
991 1305 4      |
992 1306 3      |   END;
993 1307 3      |
994 1308 3      |   $xpo_close (iob = stciob, options = remember);
995 1309 3      |   END;
996 1310 2      |
997 1311 1      |   END;
```

```
                                .PSECT $SPLITS, NOWRT, NOEXE, 2
                                00000000 0006C P.ABH: .LONG 0
00000000G 00000000G 00000000G 00070 .ADDRESS TMS_EX, TMS_FG, TMS_TB
                                0007C P.ABI: .BLKB 0
                                5D 35 31 61 76 5B 0007C P.ABJ: .ASCII \[va15]\
                                2A 6E 30 63 2A 00082 P.ABM: .ASCII \*c0n*\
                                40 40 00087 P.ABN: .ASCII \aa\
                                00089 .BLKB 3
                                00000000 0008C P.ABQ: .LONG 0
00000000G 00000000G 00000000G 00090 .ADDRESS RNO_EX, RNO_FG, RNO_TB
37 20 45 47 41 50 20 54 53 45 54 2E 0009C P.ABR: .ASCII \.TEST PAGE 7\
                                32 4B 53 2E 000A8 P.ABS: .ASCII \.SK2\
                                31 4B 53 2E 000AC P.ABT: .ASCII \.SK1\
                                2D 49 2E 000B0 P.ABV: .ASCII \.I-\
                                4B 4E 41 4C 42 2E 000B3 P.ABX: .ASCII \.BLANK\
                                000B9 .BLKB 3
00 00 00 2A 6E 31 63 2A 000BC P.ABY: .ASCII \*c1n*\<0><0><0>
                                00 00 00 2D 000C4 P.ABZ: .ASCII \-\<0><0><0>
```

```
                                .PSECT $OWNS, NOEXE, 2
                                0000 00090 $STR$STRING:
                                .WORD 0
                                01 0E 00092 .BYTE 14, 1
00000000* 00094 .ADDRESS P.ABI
                                0006 00098 $STR$STRING:
                                .WORD 6
                                01 0E 0009A .BYTE 14, 1
00000000* 0009C .ADDRESS P.ABJ
                                0005 000A0 $STR$STRINGO:
                                .WORD 5
```



```
01 0E 000A2 .BYTE 14, 1
00000000' 000A4 .ADDRESS P.ABM
0002 000AB $STR$STRING2:
01 0E 000AA .WORD 2
00000000' 000AC .BYTE 14, 1
000C 000B0 .ADDRESS P.ABN
0000 000B0 $STR$STRING:
01 0E 000B2 .WORD 12
00000000' 000B4 .BYTE 14, 1
0004 000B8 .ADDRESS P.ABR
0000 000B8 $STR$STRING:
01 0E 000BA .WORD 4
00000000' 000BC .BYTE 14, 1
0004 000C0 .ADDRESS P.ABS
0000 000C0 $STR$STRING:
01 0E 000C2 .WORD 4
00000000' 000C4 .BYTE 14, 1
0003 000C8 .ADDRESS P.ABT
0000 000C8 $STR$STRING0:
01 0E 000CA .WORD 3
00000000' 000CC .BYTE 14, 1
0006 000D0 .ADDRESS P.ABV
0000 000D0 $STR$STRING:
01 0E 000D2 .WORD 6
00000000' 000D4 .BYTE 14, 1
0000 000D4 .ADDRESS P.ABX
```

TMS_TITLES= P.ABH
RNO_TITLES= P.ABQ

```
.PSECT $CODE$,NOWRT,2
OFFC 00000
.ENTRY STCHDR, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-
R11
5B 00000000G EF 9E 00002 MOVAB IOB$+68, R11
5A 00000000G EF 9E 00009 MOVAB $STR$TARGET, R10
59 00000000G EF 9E 00010 MOVAB IOB$+52, R9
5E 18 C2 00017 SUBL2 #24, SP
7FFFFFFF 8F 00000000G EF D1 0001A 1$: CMPL NXTSTC, #2147483647
06 13 00025 BEQL 2$
00000000G EF D5 00027 TSTL CMDBLK+4
01 12 0002D 2$: BNEQ 3$
04 0002F RET
58 01 D0 00030 3$: MOVL #1, FIRST RECORD
56 00000000G EF D0 00033 MOVL NXTSTC, STCN
00000000G EF 7FFFFFFF 8F D0 0003A MOVL #2147483647, NXTSTC
FA A9 00010001 8F C8 00045 BISL2 #65537, IOB$+46
FB A9 01 90 0004D MOVAB #1, IOB$+44
00000000G EF 9F 00051 PUSHAB OPEN_ERROR
7E D4 00057 CLRL -(SP)
CC A9 9F 00059 PUSHAB IOB$
00000000G EF 03 FB 0005C CALLS #3, XPOSOPEN
69 08 B0 00063 4$: MOVW #8, IOB$+52
02 A9 02 90 00066 MOVAB #2, IOB$+54
FB A9 06 90 0006A MOVAB #6, IOB$+44
00000000G EF 9F 0006E PUSHAB XPOSFAILURE
```

00000000G	EF	CC	7E	D4	00074	CLRL	-(SP)	
00208001	8F		A9	9F	00076	PUSHAB	IOB\$	
			03	FB	00079	CALLS	#3, XPOSGET	
			50	D1	00080	CMPL	R0, #2129921	
			03	13	00087	BEQL	5\$	
			0715	31	00089	BRW	52\$	
00000000G	50	04	A9	D0	0008C	5\$:	MOVL	STCIOB+56, R0
00000000G	EF		60	D0	00090		MOVL	(R0), MAJOR
00000000G	EF	04	A0	D0	00097		MOVL	4(R0), MINOR
			51	D4	0009F		CLRL	R1
02 00000000G			EF	D1	000A1		CMPL	MAJOR, #2
			02	12	000A8		BNEQ	6\$
			51	D6	000AA		INCL	R1
			50	D4	000AC	6\$:	CLRL	R0
01 00000000G			EF	D1	000AE		CMPL	MINOR, #1
			02	12	000B5		BNEQ	7\$
			50	D6	000B7		INCL	R0
52			51	D2	000B9	7\$:	MCOML	R1, R2
50			52	CA	000BC		BICL2	R2, R0
01			50	D1	000BF		CMPL	R0, #1
			30	12	000C2		BNEQ	10\$
02 69			10	B0	000C4		MOVW	#16, IOB\$+52
F8 A9			02	90	000C7		MOVB	#2, IOB\$+54
			06	90	000CB		MOVB	#6, IOB\$+44
00000000G			EF	9F	000CF		PUSHAB	XPOSFAILURE
			7E	D4	000D5		CLRL	-(SP)
			A9	9F	000D7		PUSHAB	IOB\$
00000000G	EF	CC	03	FB	000DA		CALLS	#3, XPOSGET
			50	D4	000E1		CLRL	I
00000000GEF40		04	B940	D0	000E3	8\$:	MOVL	@STCIOB+56[I], PAGEN[I]
F2 50			03	F3	000ED		AOBLEQ	#3, I, 8\$
			FF6F	31	000F1	9\$:	BRW	4\$
			52	D4	000F4	10\$:	CLRL	R2
03 00000000G			EF	D1	000F6		CMPL	MAJOR, #3
			02	12	000FD		BNEQ	11\$
			52	D6	0C0FF		INCL	R2
50 00000000G			EF	D0	00101	11\$:	MOVL	MINOR, R0
			03	18	00108		BGEQ	12\$
50			50	CE	0010A		MNEGL	R0, R0
			51	D4	0010D	12\$:	CLRL	R1
56			50	D1	0010F		CMPL	R0, STCN
			02	13	00112		BEQL	13\$
			51	D6	00114		INCL	R1
53			52	D2	00116	13\$:	MCOML	R2, R3
51			53	CA	00119		BICL2	R3, R1
01			51	D1	0011C		CMPL	R1, #1
			72	12	0011F		BNEQ	17\$
50 00000000G			EF	D0	00121		MOVL	MINOR, R0
			03	18	00128		BGEQ	14\$
50			50	CE	0012A		MNEGL	R0, R0
56			50	D1	0012D	14\$:	CMPL	R0, STCN
			17	15	00130		BLEQ	16\$
00000000G	EF		50	D1	00132		CMPL	R0, NXTSTC
			07	15	00139		BLEQ	15\$
50 00000000G			EF	D0	0013B		MOVL	NXTSTC, R0
00000000G	EF		50	D0	00142	15\$:	MOVL	R0, NXTSTC
69			04	B0	00149	16\$:	MOVW	#4, IOB\$+52

1065
1067
1068
1073

1078

1087
10881070
1093

1100

1105

02	A9	02	90	0014C	MOVB	#2, IOB\$+54		
F8	A9	06	90	00150	MOVB	#6, IOB\$+44		
		EF	9F	00154	PUSHAB	XPOSFAILURE		
		7E	D4	0015A	CLRL	-(SP)		
		A9	9F	0015C	PUSHAB	IOB\$		
00000000G	EF	03	FB	0015F	CALLS	#3, XPOSGET		
	50	B9	D0	00166	MOVL	@STC IOB+56, R0	1107	
		85	13	0016A	BEQL	9\$		
	50	03	C0	0016C	ADDL2	#3, R0	1110	
	50	04	C6	0016F	DIVL2	#4, R0		
69	50	04	A5	00172	MULW3	#4, R0, IOB\$+52		
	50	02	90	00176	MOVB	#2, IOB\$+54		
	F8	06	90	0017A	MOVB	#6, IOB\$+44		
	A9	EF	9F	0017E	PUSHAB	XPOSFAILURE		
		7E	D4	00184	CLRL	-(SP)		
		A9	9F	00186	PUSHAB	IOB\$		
00000000G	EF	03	FB	00189	CALLS	#3, XPOSGET		
		FED0	31	00190	BRW	4\$	1070	
		50	D4	00193	CLRL	R0	1114	
	56	EF	D1	00195	CMPL	MINOR, STCN		
		02	12	0019C	BNEQ	18\$		
		50	D6	0019E	INCL	R0		
	51	52	D2	001A0	MCOML	R2, R1		
	50	51	CA	001A3	BICL2	R1, R0		
	01	50	D1	001A6	CMPL	R0, #1		
		03	13	001A9	BEQL	19\$		
		0583	31	001AB	BRW	49\$		
00000000G	EF	EF	9E	001AE	MOVAB	LINE, LP	1123	
		00000000G	EF	D4	001B9	CLRL	INTLIN	
		00000000G	EF	D4	001BF	CLRL	EXTLIN	
00000000G	EF	00000000G	EF	D0	001C5	MOVL	INDENTS, LINE INDENT	1124
00000000G	EF	00000000G	EF	D0	001D0	MOVL	LINE INDENT, EXTLIN	1125
00000000G	EF	00000000G	EF	C1	001DB	ADDL3	INDENTS+4, INDENTS, WRAP	1126
00000000G	EF	00000000G	EF	D0	001EB	MOVL	CMDBLK+24, RMARGIN	1127
00000000G	EF		02	D0	001F6	MOVL	#2, MAJOR	1128
	03		58	E8	001FD	BLBS	FIRST_RECORD, 21\$	1130
		029A	31	00200	BRW	34\$		
	03	56	D1	00203	CMPL	STCN, #3		
		F8	14	00206	BGTR	20\$		
52		02	78	00208	ASHL	#2, STCN, R2	1147	
03	00000000G	01	E0	0020C	BBS	#1, CMDBLK, 22\$	1136	
		00C9	31	00214	BRW	25\$		
		00000000G	EF	9F	00217	PUSHAB	STR\$FAILURE	1143
			7E	D4	0021D	CLRL	-(SP)	
			5A	DD	0021F	PUSHL	R10	
		00000000'	EF	9F	00221	PUSHAB	\$STR\$STRING	
			7E	D4	00227	CLRL	-(SP)	
00000000G	EF	05	FB	00229	CALLS	#5, XST\$COPY		
	50	6A	3C	00230	MOVZWL	TMPSTR, R0		
00000000G	EF	50	C0	00233	ADDL2	R0, CHROUT		
	6B	6A	9E	0023A	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
E8	AB	07	90	0023D	MOVB	#7, IOB\$+44		
		00000000G	EF	9F	00241	PUSHAB	XPOSFAILURE	
			7E	D4	00247	CLRL	-(SP)	
		BC	AB	9F	00249	PUSHAB	IOB\$	
00000000G	EF	03	FB	0024C	CALLS	#3, XPO\$PUT		
07	00000000G	01	E1	00253	BBC	#1, CMDBLK, 23\$		

00000000G	EF	00000000G	00	FB	0025B	CALLS	#0, SPLIT		
			EF	D5	00262	TSTL	HL N		1145
		00000000G	4B	13	00268	BEQL	24\$		
			EF	9F	0026A	PUSHAB	STR\$FAILURE		
			7E	D4	00270	CLRL	-(SP)		
		00000000'	5A	DD	00272	PUSHL	R10		
			EF	9F	00274	PUSHAB	\$STR\$STRING		
			7E	D4	0027A	CLRL	-(SP)		
00000000G	EF		05	FB	0027C	CALLS	#5, XST\$COPY		
00000000G	50		6A	3C	00283	MOVZWL	TMPSTR, R0		
	EF		50	C0	00286	ADDL2	R0, CHROUT		
	6B		6A	9E	0028D	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
E8	AB		07	90	00290	MOVB	#7, IOB\$+44		
		00000000G	EF	9F	00294	PUSHAB	XPOS\$FAILURE		
			7E	D4	0029A	CLRL	-(SP)		
		BC	AB	9F	0029C	PUSHAB	IOB\$		
00000000G	EF		03	FB	0029F	CALLS	#3, XPOS\$PUT		
07	EF		01	E1	002A6	BBC	#1, CMDBLK, 24\$		
00000000G	EF		00	FB	002AE	CALLS	#0, SPLIT		
		00000000'	EF	9F	002B5	PUSHAB	\$STR\$STRING2		1147
		00000000'	EF	9F	002BB	PUSHAB	TMS TITLES[R2]		
			9E	DD	002C2	PUSHL	@(SP)+		
		00000000'	EF	9F	002C4	PUSHAB	\$STR\$STRINGO		
00000000G	EF		03	FB	002CA	CALLS	#3, XST\$JOIN		
		00000000G	EF	9F	002D1	PUSHAB	STR\$FAILURE		
			7E	D4	002D7	CLRL	-(SP)		
		0401	8F	BB	002D9	PUSHR	#*M<R0,R10>		
		0180	31	002DD	BRW	32\$			
		00000000'	EF	9F	002E0	PUSHAB	RNO TITLES[R2]		1154
	52		9E	DD	002E7	MOVL	@(SP)+ R2		
		00000000G	EF	9F	002EA	PUSHAB	STR\$FAILURE		1156
			7E	D4	002F0	CLRL	-(SP)		
			5A	DD	002F2	PUSHL	R10		
		00000000'	EF	9F	002F4	PUSHAB	\$STR\$STRING		
			7E	D4	002FA	CLRL	-(SP)		
00000000G	EF		05	FB	002FC	CALLS	#5, XST\$COPY		
	50		6A	3C	00303	MOVZWL	TMPSTR, R0		
00000000G	EF		50	C0	00306	ADDL2	R0, CHROUT		
	6B		6A	9E	0030D	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
E8	AB		07	90	00310	MOVB	#7, IOB\$+44		
		00000000G	EF	9F	00314	PUSHAB	XPOS\$FAILURE		
			7E	D4	0031A	CLRL	-(SP)		
		BC	AB	9F	0031C	PUSHAB	IOB\$		
00000000G	EF		03	FB	0031F	CALLS	#3, XPOS\$PUT		
07	EF		01	E1	00326	BBC	#1, CMDBLK, 26\$		
00000000G	EF		00	FB	0032E	CALLS	#0, SPLIT		
		00000000G	EF	D5	00335	TSTL	HL N		1157
			12	13	0033B	BEQL	27\$		
		00000000G	EF	9F	0033D	PUSHAB	STR\$FAILURE		
			7E	D4	00343	CLRL	-(SP)		
			5A	DD	00345	PUSHL	R10		
		00000000'	EF	9F	00347	PUSHAB	\$STR\$STRING		
			10	11	0034D	BRB	28\$		
		00000000G	EF	9F	0034F	PUSHAB	STR\$FAILURE		
			7E	D4	00355	CLRL	-(SP)		
			5A	DD	00357	PUSHL	R10		
		00000000'	EF	9F	00359	PUSHAB	\$STR\$STRING		


```
00000000G EF 7E D4 0035F 28$: CLRL -(SP)
00000000G 50 05 FB 00361 CALLS #5, XST$COPY
00000000G EF 6A 3C 00368 MOVZWL TMPSTR, R0
E8 AB 50 C0 00368 ADDL2 R0, CHROUT
00000000G 6A 9E 00372 MOVAB $IOB$OUTPUT, IOB$+68
00000000G EF 07 90 00375 MOVB #7, IOB$+44
00000000G 00000000G EF 9F 00379 PUSHAB XPOS$FAILURE
BC 7E D4 0037F CLRL -(SP)
00000000G AB 9F 00381 PUSHAB IOB$
00000000G EF 03 FB 00384 CALLS #3, XPOS$PUT
00000000G EF 01 E1 0038B BBC #1, CMDBLK, 29$
00000000G EF 00 FB 00393 CALLS #0, SPLIT
00000000G 50 00000000G EF D0 0039A 29$: MOVL INDENTS, R0
66 13 003A1 BEQL 30$
7E D4 003A3 CLRL -(SP)
50 DD 003A5 PUSHL R0
00000000G 7E 0903 8F 3C 003A7 MOVZWL #2307, -(SP)
EF 03 FB 003AC CALLS #3, XST$ASCII
50 DD 003B3 PUSHL R0
00000000G EF 00000000' EF 9F 003B5 PUSHAB $STR$STRINGO
00000000G EF 00000000G EF 02 FB 003BB CALLS #2, XST$JOIN
00000000G EF 9F 003C2 PUSHAB STR$FAILURE
0401 7E D4 003C8 CLRL -(SP)
8F BB 003CA PUSHR #*M<R0,R10>
7E D4 003CE CLRL -(SP)
00000000G EF 05 FB 003D0 CALLS #5, XST$COPY
50 6A 3C 003D7 MOVZWL TMPSTR, R0
00000000G EF 50 C0 003DA ADDL2 R0, CHROUT
E8 AB 6A 9E 003E1 MOVAB $IOB$OUTPUT, IOB$+68
00000000G 07 90 003E4 MOVB #7, IOB$+44
00000000G EF 9F 003E8 PUSHAB XPOS$FAILURE
BC 7E D4 003EE CLRL -(SP)
AB 9F 003F0 PUSHAB IOB$
00000000G EF 03 FB 003F3 CALLS #3, XPOS$PUT
00000000G EF 01 E1 003FA BBC #1, CMDBLK, 30$
00000000G EF 00 FB 00402 CALLS #0, SPLIT
00000000G 00000000G EF 9F 00409 30$: PUSHAB STR$FAILURE
0404 7E D4 0040F CLRL -(SP)
8F BB 00411 PUSHR #*M<R2,R10>
7E D4 00415 CLRL -(SP)
00000000G EF 05 FB 00417 CALLS #5, XST$COPY
50 6A 3C 0041E MOVZWL TMPSTR, R0
00000000G EF 50 C0 00421 ADDL2 R0, CHROUT
E8 AB 6A 9E 00428 MOVAB $IOB$OUTPUT, IOB$+68
00000000G 07 90 0042B MOVB #7, IOB$+44
00000000G EF 9F 0042F PUSHAB XPOS$FAILURE
BC 7E D4 00435 CLRL -(SP)
AB 9F 00437 PUSHAB IOB$
00000000G EF 03 FB 0043A CALLS #3, XPOS$PUT
00000000G EF 01 E1 00441 BBC #1, CMDBLK, 31$
00000000G EF 00 FB 00449 CALLS #0, SPLIT
00000000G 00000000G EF 9F 00450 31$: PUSHAB STR$FAILURE
7E D4 00456 CLRL -(SP)
5A DD 00458 PUSHL R10
00000000' EF 9F 0045A PUSHAB $STR$STRING
00000000G EF 7E D4 00460 32$: CLRL -(SP)
05 FB 00462 CALLS #5, XST$COPY
```

1159

1161

1163

1164

CAPTION
V04-000

CAPTION - Generate figure, table and example ca
STCHDR - Process the .SEND TOC n captions temp

G 12

15-Sep-1984 23:50:23
14-Sep-1984 13:05:37

VAX-11 Bliiss-32 V4.0-742
[RUNOFF.SRC]CAPTION.BLI;1

Page 39
(3)

00000000G	50	6A	3C	00469	MOVZWL	TMPSTR, R0	
	EF	50	C0	0046C	ADDL2	R0, CHROUT	
E8	6B	6A	9E	00473	MOVAB	\$IOR\$OUTPUT, 10B\$+68	
	AB	07	90	00476	MOVB	#7, 10B\$+44	
00000000G		EF	9F	0047A	PUSHAB	XPOSFAILURE	
	BC	7E	D4	00480	CLRL	-(SP)	
		AB	9F	00482	PUSHAB	10B\$	
00000000G	EF	03	FB	00485	CALLS	#3, XPOSPUT	
07	EF	01	E1	0048C	BBC	#1, CMDBLK, 33\$	
00000000G	EF	00	FB	00494	CALLS	#0, SPLIT	
		58	D4	0049B	CLRL	FIRST RECORD	1167
00000000G	EF	01	D0	0049D	MOVL	#1, HE N	1170
1A	EF	01	E1	004A4	BBC	#1, CMDBLK, 35\$	1172
00000000G	EF	05	28	004AC	MOV C3	#5, P.ABY, ALP	
FF	EF	05	C0	004B8	ADDL2	#5, LP	
00000000G	EF	05	C0	004BF	ADDL2	#5, INTLIN	
	69	04	B0	004C6	MOVW	#4, 10B\$+52	1177
02	A9	02	90	004C9	MOVB	#2, 10B\$+54	
F8	A9	06	90	004CD	MOVB	#6, 10B\$+44	
00000000G		EF	9F	004D1	PUSHAB	XPOSFAILURE	
	CC	7E	D4	004D7	CLRL	-(SP)	
		A9	9F	004D9	PUSHAB	10B\$	
00000000G	EF	03	FB	004DC	CALLS	#3, XPOSGET	
OC	AE	04	B9	004E3	MOVL	@STC10B+56, LEN	1178
50	AE	03	C1	004E8	ADDL3	#3, LEN, R0	1179
	50	04	C6	004ED	DIVL2	#4, R0	
69	50	04	A5	004F0	MULW3	#4, R0, 10B\$+52	
	02	02	90	004F4	MOVB	#2, 10B\$+54	
F8	A9	06	90	004F8	MOVB	#6, 10B\$+44	
00000000G		EF	9F	004FC	PUSHAB	XPOSFAILURE	
	CC	7E	D4	00502	CLRL	-(SP)	
		A9	9F	00504	PUSHAB	10B\$	
00000000G	EF	03	FB	00507	CALLS	#3, XPOSGET	
04	OC	2C	3A	0050E	LOCC	#4, LEN, @STC10B+56	1180
		02	12	00514	BNEQ	36\$	
	08	51	D4	00516	CLRL	R1	
03	AE	51	D0	00518	MOVL	R1, PTR	
00000000G	EF	04	E0	0051C	BBS	#4, CMDBLK, 38\$	1182
		009A	31	00524	BRW	40\$	
		FB	13	00527	BEQL	37\$	1188
	04	A9	D0	00529	MOVL	STC10B+56, SEC_P	1198
04	AE	6E	C3	0052D	SUBL3	SEC_P, PTR, SEC_L	1199
		5E	DD	00533	PUSHL	SP	1200
	08	AE	9F	00535	PUSHAB	SEC_L	
00000000V	EF	02	FB	00538	CALLS	#2, STRIP	
		04	AE	D5	TSTL	SEC_L	1202
		7D	13	00542	BEQL	40\$	
	OF	EF	93	00544	BITB	PAGEN, #15	1208
		47	13	0054B	BEQL	39\$	
00000000G	EF	EF	9E	0054D	MOVAB	TXTSCT, PTRSCT	1214
		EF	9F	00558	PUSHAB	PTRSCT	1215
		EF	9F	0055E	PUSHAB	PAGEN	
00000000G	EF	02	FB	00564	CALLS	#2, PACSEC	
00000000G	EF	50	D0	0056B	MOVL	R0, LENSCT	
		EF	9F	00572	PUSHAB	TXTSCT	1216
		EF	DD	00578	PUSHL	LENSCT	
00000000G	EF	02	FB	0057E	CALLS	#2, FMTXT	

CAPTION
V04-000

CAPTION - Generate figure, table and example ca 15-Sep-1984 23:50:23
STCHDR - Process the .SEND TOC n captions temp 14-Sep-1984 13:05:37

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]CAPTION.BLI;1

Page 40
(3)

57

			00000000'	EF	9F	00585	PUSHAB	P,ABZ	1217
				01	DD	0058B	PUSHL	#1	
	00000000G	EF		02	FB	0058D	CALLS	#2, FMTTXT	
			08	6E	DD	00594	PUSHL	SEC_P	1220
				AE	DD	00596	PUSHL	SEC_L	
21	00000000G	EF		02	FB	00599	CALLS	#2, FMTTXT	
	00000000G	EF		01	EO	005A0	BBS	#1, CMDBLK, 41\$	1222
	00000000G	FF		20	90	005A8	MOVB	#32, @LP	1224
			00000000G	EF	D6	005AF	INCL	LP	
	00000000G	EF		EF	D6	005B5	INCL	INTLIN	
	00000000G	EF		EF	D6	005BB	INCL	EXTLIN	
16	00000000G	EF		01	E1	005C1	BBC	#1, CMDBLK, 42\$	1232
	00000000G	FF	40	8F	90	005C9	MOVB	#64, @LP	1234
			00000000G	EF	D6	005D1	INCL	LP	
			00000000G	EF	D6	005D7	INCL	INTLIN	
				3B	11	005DD	BRB	43\$	1232
57	00000000G	EF	00000000G	EF	C1	005DF	ADDL3	INDENTS+4, INDENTS, R7	1236
	00000000G	EF		57	D1	005EB	CMPL	R7, EXTLIN	
				26	15	005F2	BLEQ	43\$	
			00000000G	EF	C2	005F4	SUBL2	EXTLIN, R7	
20		6E		00	2C	005FB	MOVCS	#0, (SP), #32, R7, @LP	
			00000000G	FF		00600			
	00000000G	EF		57	C0	00605	ADDL2	R7, LP	
	00000000G	EF		57	C0	0060C	ADDL2	R7, INTLIN	
	00000000G	EF		57	C0	00613	ADDL2	R7, EXTLIN	
	00000000G	EF	00000000G	EF	9E	0061A	MOVAB	TXIPAG, PTRPAG	1241
18	00000000G	EF		03	EO	00625	BBS	#3, CMDBLK, 44\$	1243
				0A	DD	0062D	PUSHL	#10	1249
			00000000G	EF	9F	0062F	PUSHAB	PTRPAG	
	00000000G	7E	00000000G	EF	3C	00635	MOVZWL	PAGEN+14, -(SP)	
				03	FB	0063C	CALLS	#3, PACBAS	
				13	11	00643	BRB	45\$	
			00000000G	EF	9F	00645	PUSHAB	PTRPAG	1256
			00000000G	EF	9F	0064B	PUSHAB	PAGEN	
	00000000G	EF		02	FB	00651	CALLS	#2, PACPAG	
	00000000G	EF		50	D0	0065B	MOVL	R0, LENPAG	
50	00000000G	EF	00000000G	EF	C3	0065F	SUBL3	LENPAG, CMDBLK+24, R0	1259
	00000000G	EF	FF	A0	9E	0066B	MOVAB	-1(R0), RMARGIN	
	00000000G	EF	00000000G	EF	D0	00673	MOVL	EXTLIN, WRAP	1260
			08	AE	D5	0067E	TSTL	PTR	1264
				07	12	00681	BNEQ	46\$	
	08	AE	04	A9	D0	00683	MOVL	STCIOB+56, PTR	1266
				DD	11	00688	BRB	47\$	
			08	AE	D6	0068A	INCL	PTR	1273
50	04	A9	08	AE	C3	0068D	SUBL3	PTR, STCIOB+56, R0	1274
	0C	AE		50	C0	00693	ADDL2	R0, LEN	
			08	AE	9F	00697	PUSHAB	PTR	1277
			10	AE	9F	0069A	PUSHAB	LEN	
	00000000V	EF		02	FB	0069D	CALLS	#2, STRIP	1278
			08	AE	DD	006A4	PUSHL	PTR	
			10	AE	DD	006A7	PUSHL	LEN	
	00000000G	EF		02	FB	006AA	CALLS	#2, FMTTXT	
	00000000G	EF		00	FB	006B1	CALLS	#0, INSREF	1282
	10	AE	00000000G	EF	B0	006B8	MOVW	INTLIN, \$STR\$STRING	1286
	12	AE		0E	90	006C0	MOVB	#14, \$STR\$STRING+2	
	13	AE		01	90	006C4	MOVB	#1, \$STR\$STRING+3	
	14	AE	00000000G	EF	9E	006C8	MOVAB	LINE, \$STR\$STRING+4	

		00000000G	EF	9F	006D0	PUSHAB	STR\$FAILURE	
			7E	D4	006D6	CLRL	-(SP)	
		1C	5A	DD	006D8	PUSHL	R10	
			AE	9F	006DA	PUSHAB	\$STR\$STRING	
			7E	D4	006DD	CLRL	-(SP)	
	00000000G	EF	05	FB	006DF	CALLS	#5, XST\$COPY	
		50	6A	3C	006E6	MOVZWL	TMPSTR, R0	
	00000000G	EF	50	C0	006E9	ADDL2	R0, CHROUT	
		6B	6A	9E	006F0	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
	E8	AB	07	90	006F3	MOVB	#7, IOB\$+44	
		00000000G	EF	9F	006F7	PUSHAB	XPO\$FAILURE	
			7E	D4	006FD	CLRL	-(SP)	
		BC	AB	9F	006FF	PUSHAB	IOB\$	
	00000000G	EF	03	FB	00702	CALLS	#3, XPO\$PUT	
07	00000000G	EF	01	E1	00709	BBC	#1, CMDBLK, 48\$	
	00000000G	EF	00	FB	00711	CALLS	#0, SPLIT	
	00000000G	EF	9E	9E	00718	MOVAB	LINE, LP	1287
		00000000G	EF	D4	00723	CLRL	INTLIN	
		00000000G	EF	D4	00729	CLRL	EXTLIN	
			6D	11	0072F	BRB	51\$	1070
	51		56	CE	00731	MNEGL	STCN, R1	1290
			50	D4	00734	CLRL	R0	
	51	00000000G	EF	D1	00736	CMPL	MINOR, R1	
			02	12	0073D	BNEQ	50\$	
			50	D6	0073F	INCL	R0	
	53		52	D2	00741	MCOML	R2, R3	
	50		53	CA	00744	BICL2	R3, R0	
	01		50	D1	00747	CMPL	R0, #1	
			52	12	0074A	BNEQ	51\$	
	69		04	B0	0074C	MOVW	#4, IOB\$+52	1298
	02		02	90	0074F	MOVB	#2, IOB\$+54	
	F8		06	90	00753	MOVB	#6, IOB\$+44	
		00000000G	EF	9F	00757	PUSHAB	XPO\$FAILURE	
			7E	D4	0075D	CLRL	-(SP)	
		CC	A9	9F	0075F	PUSHAB	IOB\$	
	00000000G	EF	03	FB	00762	CALLS	#3, XPO\$GET	
		04	B9	D0	00769	MOVL	\$STCIOB+56, LEN	1299
		03	A2	9E	0076D	MOVAB	3(R2), R0	1300
			04	C6	00771	DIVL2	#4, R0	
69			04	A5	00774	MULW3	#4, R0, IOB\$+52	
	02		02	90	00778	MOVB	#2, IOB\$+54	
	F8		06	90	0077C	MOVB	#6, IOB\$+44	
		00000000G	EF	9F	00780	PUSHAB	XPO\$FAILURE	
			7E	D4	00786	CLRL	-(SP)	
		CC	A9	9F	00788	PUSHAB	IOB\$	
	00000000G	EF	03	FB	0078B	CALLS	#3, XPO\$GET	
		04	A9	DD	00792	PUSHL	STCIOB+56	1301
			52	DD	00795	PUSHL	LEN	
	00000000V	EF	02	FB	00797	CALLS	#2, INSRNT	
			F8C2	31	0079E	BRW	4\$	1059
	FA		10	88	007A1	BISB2	#16, IOB\$+46	1308
	F8		02	90	007A5	MOVB	#2, IOB\$+44	
		00000000G	EF	9F	007A9	PUSHAB	XPO\$FAILURE	
			7E	D4	007AF	CLRL	-(SP)	
		CC	A9	9F	007B1	PUSHAB	IOB\$	
	00000000G	EF	03	FB	007B4	CALLS	#3, XPO\$CLOSE	
			F85C	31	007BB	BRW	1\$	1045

CAPTION
V04-000

CAPTION - Generate figure, table and example ca 15-Sep-1984 23:50:23
STCHDR - Process the .SEND TOC n captions temp 14-Sep-1984 13:05:37

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]CAPTION.BLI;1

Page 42
(3)

04 007BE

RET

; 1311

; Routine Size: 1983 bytes, Routine Base: \$CODE\$ + 0D70

CAPTION
V04-000

CAPTION
STRIP -

- Generate figure, table and example ca 15-Sep-1984 23:50:23
Strip leading and trailing whitespace f 14-Sep-1984 13:05:37

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]CAPTION.BLI;1

Page 43
(4)

```
999 1312 1 %SBTTL 'STRIP - Strip leading and trailing whitespace from text'
1000 1313 1 ROUTINE strip (l_addr, p_addr) : NOVALUE =
1001 1314 1 ++
1002 1315 1
1003 1316 1 FUNCTIONAL DESCRIPTION:
1004 1317 1
1005 1318 1 This routine strips leading and trailing whitespace from the input string
1006 1319 1
1007 1320 1 FORMAL PARAMETERS:
1008 1321 1
1009 1322 1 l_addr - address of length of string
1010 1323 1 p_addr - address of CHSPTR to string
1011 1324 1
1012 1325 1 IMPLICIT INPUTS:
1013 1326 1
1014 1327 1 None
1015 1328 1
1016 1329 1 IMPLICIT OUTPUTS:
1017 1330 1
1018 1331 1 The length and string pointer are modified to reflect stripped whitespace
1019 1332 1
1020 1333 1 ROUTINE VALUE:
1021 1334 1 COMPLETION CODES:
1022 1335 1
1023 1336 1 None
1024 1337 1
1025 1338 1 SIDE EFFECTS:
1026 1339 1
1027 1340 1 None
1028 1341 1
1029 1342 1 --
1030 1343 1
1031 1344 1 BEGIN
1032 1345 1
1033 1346 1 LOCAL
1034 1347 1 len,
1035 1348 1 ptr;
1036 1349 1
1037 1350 1 len = ..l_addr;
1038 1351 1 ptr = ..p_addr;
1039 1352 1
1040 1353 1 DECR i FROM .len TO 1 DO
1041 1354 1 BEGIN
1042 1355 1
1043 1356 1 Remove leading whitespace from section number
1044 1357 1
1045 1358 1 LOCAL
1046 1359 1 ch;
1047 1360 1
1048 1361 1 ch = CH$RCHAR (.ptr);
1049 1362 1
1050 1363 1 IF (.ch EQL %C' ') OR (.ch EQL tab)
1051 1364 1 THEN
1052 1365 1 BEGIN
1053 1366 1 len = .len - 1;
1054 1367 1 ptr = CH$PLUS (.ptr, 1);
1055 1368 1 END
```

```
1056 1369 ELSE
1057 1370 EXITLOOP;
1058 1371
1059 1372 END;
1060 1373
1061 1374 .p_addr = .ptr; ! Set pointer to beginning of text
1062 1375 ptr = CH$PLUS (.ptr, .len - 1); ! Point to last character
1063 1376
1064 1377 DECR 1 FROM .len TO 1 DO
1065 1378 BEGIN
1066 1379 ! Remove trailing blanks and tabs
1067 1380
1068 1381 LOCAL
1069 1382 ch;
1070 1383
1071 1384 ch = CH$RCHAR (.ptr);
1072 1385
1073 1386 IF (.ch EQL %C' ') OR (.ch EQL tab)
1074 1387 THEN
1075 1388 BEGIN
1076 1389 len = .len - 1;
1077 1390 ptr = CH$PLUS (.ptr, -1);
1078 1391
1079 1392 END
1080 1393 ELSE
1081 1394 EXITLOOP;
1082 1395
1083 1396 END;
1084 1397
1085 1398 WHILE .len GEQ 3 DO
1086 1399 BEGIN
1087 1400 ! Strip trailing emphasis
1088 1401
1089 1402 IF CH$RCHAR (CH$PLUS (.ptr, -2)) EQL rintes
1090 1403 THEN
1091 1404 BEGIN
1092 1405 len = .len - 3;
1093 1406 ptr = CH$PLUS (.ptr, -3);
1094 1407
1095 1408 END
1096 1409 ELSE
1097 1410 EXITLOOP;
1098 1411
1099 1412 END;
1100 1413
1101 1414 .l_addr = .len; ! Return length of string
1102 1415 END;
```

```
50 04 000C 00000 STRIP: .WORD Save R2,R3
52 08 BC D0 00002 MOVL @L_ADDR, LEN
51 01 BC D0 00006 MOVL @P_ADDR, PTR
AO 9E 0000A MOVAB 1(R0), 1
11 11 0000E BRB 3$
```

```
1313
1350
1351
1353
```

CAPTION
V04-000

M 12
CAPTION - Generate figure, table and example ca 15-Sep-1984 23:50:23
STRIP - Strip leading and trailing whitespace f 14-Sep-1984 13:05:37

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]CAPTION.BLI;1

Page 45
(4)

53	62	9A	00010	1\$:	MOVZBL	(PTR), CH	1361	
20	53	D1	00013		CMPL	CH, #32	1363	
	05	13	00016		BEQL	2\$		
09	53	D1	00018		CMPL	CH, #9		
	07	12	0001B		BNEQ	4\$		
	50	D7	0001D	2\$:	DECL	LEN	1366	
	52	D6	0001F		INCL	PTR	1367	
08	EC	51	F5	00021	3\$:	SOBGTR	I, 1\$	1353
BC	52	42	D0	00024	4\$:	MOVL	PTR, @P_ADDR	1374
52	FF	A0	9E	00028		MOVAB	-1(LEN)[PTR], PTR	1375
51	01	A0	9E	0002D		MOVAB	1(R0), I	1377
	11	11	00031		BRB	7\$		
53	62	9A	00033	5\$:	MOVZBL	(PTR), CH	1385	
20	53	D1	00036		CMPL	CH, #32	1387	
	05	13	00039		BEQL	6\$		
09	53	D1	0003B		CMPL	CH, #9		
	07	12	0003E		BNEQ	8\$		
	50	D7	00040	6\$:	DECL	LEN	1390	
	52	D7	00042		DECL	PTR	1391	
EC	51	F5	00044	7\$:	SOBGTR	I, 5\$	1377	
03	50	D1	00047	8\$:	CMPL	LEN, #3	1398	
	0E	19	0004A		BLSS	9\$		
1C	FE	A2	91	0004C		CMPL	-2(PTR), #28	1403
	08	12	00050		BNEQ	9\$		
50	03	C2	00052		SUBL2	#3, LEN	1406	
52	03	C2	00055		SUBL2	#3, PTR	1407	
	ED	11	00058		BRB	8\$	1403	
04	BC	50	D0	0005A	9\$:	MOVL	LEN, @L_ADDR	1414
	04	0005E			RET		1415	

; Routine Size: 95 bytes, Routine Base: \$CODE\$ + 152F


```
1104 1416 1 %SBTTL 'INSRNT - Insert text directly into .RNT file'
1105 1417 1 GLOBAL ROUTINE insrnt (len, ptr) : NOVALUE =
1106 1418 1 ++
1107 1419 1
1108 1420 1 FUNCTIONAL DESCRIPTION:
1109 1421 1
1110 1422 1 This routine inserts the given text directly into the .RNT file
1111 1423 1
1112 1424 1 FORMAL PARAMETERS:
1113 1425 1
1114 1426 1 len - length of text
1115 1427 1 ptr - CH$PTR to text
1116 1428 1
1117 1429 1 IMPLICIT INPUTS:
1118 1430 1
1119 1431 1 cmdblk - command line information block
1120 1432 1
1121 1433 1 IMPLICIT OUTPUTS:
1122 1434 1
1123 1435 1 None
1124 1436 1
1125 1437 1 ROUTINE VALUE:
1126 1438 1 COMPLETION CODES:
1127 1439 1
1128 1440 1 None
1129 1441 1
1130 1442 1 SIDE EFFECTS:
1131 1443 1
1132 1444 1 None
1133 1445 1
1134 1446 1 --
1135 1447 1
1136 1448 1 BEGIN
1137 1449 1
1138 1450 1 IF .len GTR 0
1139 1451 1 THEN
1140 1452 1 BEGIN
1141 1453 1
1142 1454 1 There is something to process
1143 1455 1
1144 1456 1 hl_n = -1;
1145 1457 1 clr_line ();
1146 1458 1 line_indent = .indents [0];
1147 1459 1 wrap = .line_indent;
1148 1460 1 extlin = .line_indent;
1149 1461 1 rmargin = .cmdblk [contents$g_page_width];
1150 1462 1
1151 1463 1 IF .cmdblk [contents$v_tms11] THEN write_char (%C'<');
1152 1464 1
1153 1465 1 fmttxt (.len, .ptr);
1154 1466 1
1155 1467 1 IF .cmdblk [contents$v_tms11] THEN write_char (%C'>');
1156 1468 1
1157 1469 1 put ((.intlin, CH$PTR (line)));
1158 1470 1 clr_line ();
1159 1471 1 END;
1160 1472 1
```

			03FC	00000	.ENTRY	INSRNT, Save R2,R3,R4,R5,R6,R7,R8,R9	1417	
	59	00000000G	EF	9E	00002	MOVAB	IOBS+68, R9	
	58	00000000G	EF	9E	00009	MOVAB	\$STR\$TARGET, R8	
	57	00000000G	EF	9E	00010	MOVAB	LINE, INDENT, R7	
	56	00000000G	EF	9E	00017	MOVAB	EXTLIN, R6	
	55	00000000G	EF	9E	0001E	MOVAB	LINE, R5	
	54	00000000G	EF	9E	00025	MOVAB	CMDBLK, R4	
	53	00000000G	EF	9E	0002C	MOVAB	INTLIN, R3	
	52	00000000G	EF	9E	00033	MOVAB	LP, R2	
	5E		08	C2	0003A	SUBL2	#8, SP	
		04	AC	D5	0003D	TSTL	LEN	1450
			01	14	00040	BGTR	1\$	
				04	00042	RET		
	00000000G	EF	01	CE	00043	MNEGL	#1, HL N	1456
	62		65	9E	0004A	MOVAB	LINE, CP	1457
			63	D4	0004D	CLRL	INTLIN	
			66	D4	0004F	CLRL	EXTLIN	
	67	20	A4	D0	00051	MOVL	INDENTS, LINE, INDENT	1458
	00000000G	EF	67	D0	00055	MOVL	LINE, INDENT, WRAP	1459
	66		67	D0	0005C	MOVL	LINE, INDENT, EXTLIN	1460
08	00000000G	EF	A4	D0	0005F	MOVL	CMDBLK+24, RMARGIN	1461
	00	B2	01	E1	00067	BBC	#1, CMDBLK, 2\$	1463
			3C	90	0006B	MOVB	#60, @LP	
			62	D6	0006F	INCL	LP	
			63	D6	00071	INCL	INTLIN	
	7E	04	AC	7D	00073	MOVQ	LEN, -(SP)	1465
08	00000000G	EF	02	FB	00077	CALLS	#2, FMTTXX	
	64		01	E1	0007E	BBC	#1, CMDBLK, 3\$	1467
	00	B2	3E	90	00082	MOVB	#62, @LP	
			62	D6	00086	INCL	LP	
			63	D6	00088	INCL	INTLIN	
	6E	04	63	B0	0008A	MOVW	INTLIN, \$STR\$STRING	1469
	02	AE	0E	90	0008D	MOVB	#14, \$STR\$STRING+2	
	03	AE	01	90	00091	MOVB	#1, \$STR\$STRING+3	
	04	AE	65	9E	00095	MOVAB	LINE, \$STR\$STRING+4	
		00000000G	EF	9F	00099	PUSHAB	STR\$FAILURE	
			7E	D4	0009F	CLRL	-(SP)	
			58	DD	000A1	PUSHL	R8	
		0C	AE	9F	000A3	PUSHAB	\$STR\$STRING	
			7E	D4	000A6	CLRL	-(SP)	
	00000000G	EF	05	FB	000A8	CALLS	#5, XST\$COPY	
	50		68	3C	000AF	MOVZWL	TMPSTR, R0	
	00000000G	EF	50	C0	000B2	ADDL2	R0, CHROUT	
	69		68	9E	000B9	MOVAB	\$IOB\$OUTPUT, IOBS+68	
	E8	A9	07	90	000BC	MOVB	#7, IOBS+44	
		00000000G	EF	9F	000C0	PUSHAB	XPO\$FAILURE	
			7E	D4	000C6	CLRL	-(SP)	
		BC	A9	9F	000C8	PUSHAB	IOBS	
	00000000G	EF	03	FB	000CB	CALLS	#3, XPO\$PUT	
07	00000000G	64	01	E1	000D2	BBC	#1, CMDBLK, 4\$	
		EF	00	FB	000D6	CALLS	#0, SPLIT	

6265 9E 000DD 4\$: MOVAB LINE LP
63 D4 000E0 CLRL INTLIN
66 D4 000E2 CLRL EXTLIN
04 000E4 RET

: 1470
:
:
: 1473

: Routine Size: 229 bytes, Routine Base: \$CODE\$ + 158E

: 1162 1474 1
: 1163 1475 1 END
: 1164 1476 0 ELUDOM

: End of module

PSECT SUMMARY

Name	Bytes	Attributes
\$SPLITS	200	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$OWNS	216	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODE\$	5747	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	156	26	252	00:00.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:CAPTION/OBJ=OBJ\$:CAPTION MSRC\$:CAPTION/UPDATE=(ENHS:CAPTION)

: Size: 5747 code + 416 data bytes
: Run Time: 02:43.9
: Elapsed Time: 04:53.0
: Lines/CPU Min: 540
: Lexemes/CPU-Min: 96060
: Memory Used: 901 pages
: Compilation Complete

0337 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY